

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Reconstrução Digital de Espaços Históricos: o caso de estudo de Mértola Virtual

Mestrado em Informática

Alexandre Antonio de Carvalho

Dissertação Orientada por:
Prof.^a Doutora Ana Paula Boler Cláudio
e Prof.^a Doutora Maria Beatriz Duarte Pereira do Carmo.

2016

Agradecimentos

Um agradecimento muito especial às minhas orientadoras, a Professora Doutora Ana Paula Boler Cláudio e a Professora Doutora Maria Beatriz Duarte Pereira do Carmo, por me terem acompanhado durante todo o percurso deste trabalho, de modo empenhado e sempre presente nas orientações. Muito obrigado!

Agradeço ao Professor Doutor Rui Filipe Antunes pelas frequentes e enriquecedoras trocas de ideias com que me apresentou ao longo deste trabalho.

Agradeço à equipa do Campo Arqueológico de Mértola, em especial a Doutora Suzana Gómes Martínez, pelas informações sobre período histórico de Mértola.

Agradeço aos Professores e Professoras do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa durante o percurso académico, pelos ensinamentos transmitidos com eficiência e dedicação.

Agradeço aos meus colegas da academia durante este percurso de formação académica e também pela convivência.

Agradeço aos funcionários e funcionárias da Faculdade de Ciências da Universidade de Lisboa, pelo excelente atendimento oferecido no decorrer deste trabalho e durante o percurso académico na instituição de ensino. O meu muito obrigado!

Ao meu querido filho João Alexandre Virmieiro de Carvalho e a minha diletta amiga e esposa Josilaine Dias Virmieiro de Carvalho, pelo apoio e motivação incondicional. Muito obrigado.

Agradeço aos meus familiares pelo incentivo aos estudos sempre em busca de novos horizontes e aprendizados, em especial à minha mãe e ao meu pai. Muito obrigado.

Agradeço ao recurso financeiro oferecido pelo governo do Brasil através do Instituto Federal de Educação, Ciência e Tecnologia do Mato Grosso – IFMT, com apoio de licença para capacitação em programa de Mestrado.

Muito obrigado!

Ao meu filho e à minha esposa.

Resumo

O principal objetivo deste trabalho foi a reconstrução digital de espaços urbanos históricos, um tema que se insere numa temática mais alargada, a herança cultural digital. Este trabalho integra-se num projeto de investigação mais amplo que envolve a simulação de multidões em cenários urbanos que recriam épocas históricas. Pretende-se que estas simulações suportem a interação do utilizador em tempo real e que possam tirar partido de algoritmos de Inteligência Artificial, motivo pelo qual foi escolhida uma plataforma usada para o desenvolvimento de video-jogos, o Unity 3D.

Neste trabalho fez-se, numa primeira fase, um levantamento do estado da arte sobre as diferentes técnicas de reconstrução tridimensional, bem como, sobre trabalhos realizados no âmbito da recriação de espaços urbanos virtuais em contexto históricos incluindo ou não a integração de humanos virtuais inteligentes.

Tendo sempre presentes duas preocupações, o baixo custo que se desejava para a solução e a necessidade de obter um modelo que suporte interação em tempo real, concebeu-se um *pipeline* para aplicar à reconstrução digital de espaços urbanos históricos.

O *pipeline* combina várias ferramentas gratuitas ou com licenças académicas e recorre à interoperabilidade de dados 3D para criar um cenário virtual. Assim, as ferramentas do *pipeline* proposto são, para cada fase, as seguintes: ArcGIS para a aquisição e processamento de informação geográfica e do relevo, com a finalidade de obter um modelo virtual preciso do terreno; AutoCAD, Blender e 3DS Max para a modelação tridimensional manual de estruturas arquitetónicas e o CityEngine para modelação procedimental e integração do cenário tridimensional completo. A utilização do CityEngine permite a geração semi-automatizada de estruturas 3D através de ficheiros de regras e a possibilidade de inserção no cenário de uma vasta diversidade de modelos tridimensionais de árvores disponíveis numa biblioteca e finalmente a sua exportação para a plataforma Unity 3D.

Ao longo de todo o processo tentou-se limitar o número de polígonos dos modelos, tentando manter a sua qualidade.

O *pipeline* foi aplicado a um caso de estudo específico: a construção do ambiente urbano virtual da vila de Mértola na época de ocupação Islâmica. Este modelo foi construído fundamentalmente com base em informação fornecida por arqueólogos do Campo Arqueológico de Mértola.

Finalmente, procedemos a um primeiro teste que prova a validade do *pipeline* proposto e do modelo produzido. O ambiente virtual construído no CityEngine foi exportado para o Unity 3D e incluiu-se um avatar no cenário. Este avatar navega no cenário de acordo com a interação do utilizador em tempo real, estando activo o mecanismo de deteção de colisões fornecido pelo motor de Física do Unity 3D. Adicionalmente, um segundo teste foi realizado por outros elementos da nossa equipa de investigação que obtiveram resposta do sistema em tempo real ao testar o cenário com grupos de personagens virtuais autónomas cujo comportamento se baseia na execução de scripts de Inteligência Artificial.

Palavras-chave: Ambiente Urbano Virtual, Reconstrução Digital de Espaços Históricos, Herança Cultural Digital, Interoperabilidade de Dados 3D, Modelação Procedimental.

Abstract

The main objective of this work was the digital reconstruction of historical urban spaces, a theme that is part of a broader subject: digital cultural heritage. This work is part of a research project involving the simulation of crowds in urban settings that recreate historical epochs. These simulations should support user interaction in real time and can make use of Artificial Intelligence algorithms. Hence, it was chosen a platform used for developing video games, Unity 3D.

In this work we did, as a first step, a survey of the state of the art concerning different three-dimensional reconstruction techniques, as well as, projects carried out in the framework of virtual recreation of urban spaces in historical contexts with or without intelligent virtual characters.

Two main concerns were always present: the low cost of the solution and the need to obtain a model that supports real time interaction. We conceived a *pipeline*, combining several free tools or with academic subscriptions, with data interoperability between them, to create the virtual scenario. The proposed pipeline resorts to the following tools: ArcGIS for the acquisition and processing of geographical information and the elevation of the terrain of our area of interest; AutoCAD, 3DS Max and Blender for 3D manual modeling of architectural elements and the CityEngine for procedural modelling and integration of the whole scene in a format that can be imported by Unity 3D. CityEngine includes a library with a wide diversity of three-dimensional models of trees that can be included in the scenario.

Throughout the whole process we tried to limit the number of polygons of the models without sacrificing their quality.

The proposed solution was applied to generate a specific virtual urban scene: the village of Mértola during the Islamic occupation. We built the 3D models based mainly on information provided by the archaeologists of Campo Arqueológico de Mértola.

By the end, we performed a first test that proves the validity of the proposed *pipeline* and of the produced model: the virtual environment produced in CityEngine was exported to the Unity 3D environment and an avatar was included in it. This avatar navigates in the 3D scene according to the user interaction performed in real time, while the collision detection mechanism provided by the Unity 3D physics engine is active. Moreover, a second test was performed by other elements of our research team. They also obtained real time while testing the scenario with groups of autonomous virtual characters running Artificial Intelligence scripts.

Keywords: Virtual Urban Environment, Digital Reconstruction of Historical Sites, Digital Cultural Heritage, 3D Data Interoperability, Procedural Modelling.

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Motivação	1
1.2	Objetivo.....	2
1.3	Contribuições	2
1.4	Organização do documento.....	4
Capítulo 2	Conceitos bases e trabalhos relacionados	5
2.1	Herança cultural digital.....	5
2.2	Técnicas de reconstrução virtual.....	7
2.2.1	Modelação manual	7
2.2.2	Fotogrametria e <i>laser scanning</i>	9
2.2.3	Modelação procedimental	12
2.2.4	Plataforma interativas.....	15
2.3	Trabalhos relacionados	16
2.3.1	Exemplos com realidade virtual.....	17
2.3.2	Exemplos com realidade aumentada.....	20
2.3.3	Exemplos a destacar	21
2.3.3.1	A catedral de St. Andrews	21
2.3.3.2	Rome Reborn	23
2.3.3.3	Castelo de Pietrabuona.....	23
2.4	Conclusão.....	24
Capítulo 3	Análise de ferramentas e caso de estudo	25
3.1	Ferramentas de software	25
3.1.1	Interoperabilidade de dados 3D.....	27
3.1.2	Modelação tridimensional	28
3.1.3	Automatização de reconstrução	29
3.1.4	Motor de renderização.....	32
3.1.5	Ferramentas adotadas	33

3.2	Caso de estudo – Mértola Virtual	35
3.3	Conclusão.....	38
Capítulo 4	<i>Pipeline</i> de geração do ambiente virtual.....	39
4.1	Geração do terreno	40
4.2	Geração de ruas e de <i>shapes</i> para os modelos arquitetónicos.....	46
4.3	Modelação tridimensional.....	49
4.3.1	Modelação tridimensional manual	50
4.3.1.1	Níveis de detalhe.....	50
4.3.1.2	Aplicação de textura	54
4.3.2	Modelação procedimental	57
4.3.2.1	Estruturas construídas com modelação procedimental	59
4.3.2.2	Estruturas construídas com modelação poligonal combinada com regras.....	67
4.4	Integração: desenvolvimento do ambiente virtual	70
4.5	Ambiente virtual interativo	73
4.5.1	Exportação para ambiente virtual interativo	73
4.5.2	Personagem virtual interativo.....	76
4.6	Discussão	77
4.7	Conclusão.....	78
Capítulo 5	Conclusões e trabalho futuro	80
5.1	Resultados obtidos	80
5.2	Trabalho futuro	83
5.3	Conclusão.....	83
Anexo A:	Documentação técnica de geração do terreno e imagem.....	85
Anexo B:	Ambiente urbano virtual	92
Anexo C:	Fases intermediária de reconstrução do ambiente urbano virtual.....	93
Anexo D:	Texturas aplicadas na modelação com regras	95
Anexo E:	Modelação 3D com LODs	98
Anexo F:	Visualização do ambiente virtual urbano no CityEngine e Unity.....	100
Anexo G:	Casas geradas com combinação de regras e modelação poligonal.....	101
Anexo H:	Princípios da carta de Londres e Sevilha.....	102
Bibliografia	104

Lista de Figura

Figura 1 – Interface de autenticação do Open Virtual Worlds.....	22
Figura 2 – Interação do utilizador pela User Interface.....	24
Figura 3 – Interoperabilidade de dados 3D.	28
Figura 4 – Ruínas atuais do bairro islâmico e planta (Torres, 2008).	36
Figura 5 – Conjunto de residências (Macias, 1996b).....	36
Figura 6 – Mértola fortificada (Armas, 1509).....	37
Figura 7 – Vista parcial de Mértola (Torres, 2008).	37
Figura 8 – Etapas para produção do ambiente urbano virtual.....	39
Figura 9 – Etapas do processamento de informação geográfica.	41
Figura 10 – Aquisição de modelo digital do terreno.....	42
Figura 11 – Aquisição de imagem de alta resolução.....	43
Figura 12 – Modelo digital do terreno processado no ArcMap.	44
Figura 13 – Imagem georreferenciada e modelo digital do terreno.	45
Figura 14 – Fluxo de atividade para recolha de dados.....	45
Figura 15 – Criação de camada terreno.....	46
Figura 16 –Imagem do ficheiro obstáculo	47
Figura 17 – Imagem de ElevationDelta do terreno.	48
Figura 18 – Visualização parcial da rua com e sem textura.....	49
Figura 19 – Visão global do pipeline modelação.....	50
Figura 20 – Contagem de polígonos para LOD.	51
Figura 21 – Modelação das casas em 3DS Max.	52
Figura 22 – Telhado com LOD 2 da casa XI, com plug-in ATiles.....	53
Figura 23 – Processo de exportação de grupo de LOD no 3DS Max.	53
Figura 24 – Geometria dentro do âmbito e a respetiva árvore de shapes.	58
Figura 25 – Regras para distribuição de vegetação 3D.....	60
Figura 26 – Inserção de torre na cena com código de regra.....	62
Figura 27 – Modelos geométricos para exportação.	63
Figura 28 –Modelo de uma torre e oresultado da sua inserção no modelo 3D.....	63
Figura 29 – Comparação entre street shape e modelo 3D da muralha.....	64
Figura 30 – Etapas de geração do modelo 3D da muralha.....	65

Figura 31 – Variadade de tipo de telhado com operadores padrões.	66
Figura 32 – Modelação poligonal com código de regra.	68
Figura 33 – Modelação poligonal com árvore de shapes.	69
Figura 34 – Modelo da casa gerada e extrato do ficheiro.	70
Figura 35 – Pipeline CityEngine.	71
Figura 36 – Processo de importação do modelo estático.	72
Figura 37 – Função alinhamento de modelos estáticos ao terreno.	73
Figura 38 – Enquadramento da plataforma de exportação do CityEngine.	74
Figura 39 – Vista do CityEngine Web Scene.	74
Figura 40 – Vista do Unity 3D do ambiente urbano virtual.	76
Figura 41 – Humano virtual no cenário.	77
Figura 42 – Visão global do pipeline proposto.	79
Figura 43 – Ambiente urbano virtual e técnica de construção.	82

Lista de Tabela

Tabela 1 – Processos de modelação aplicado a reconstrução por modelação raiz..	8
Tabela 2 – Processos de modelação aplicado a reconstrução por fotogrametria. .	10
Tabela 3 – Trabalho relacionado ao processo de laser scanning.	12
Tabela 4 – Categorização por especificidade de modelling procedural.....	13
Tabela 5 – Principais contributos para o CityEngine.....	14
Tabela 6 – Comparativo entre as plataformas OpenSim e Secondlife.....	16
Tabela 7 – Relação de trabalhos inerente a realidade virtual.....	17
Tabela 8 – Características específica em Humano virtual.	20
Tabela 9 – Herança cultural com realidade aumentada.	20
Tabela 10 – Características técnica do OpenSim.....	22
Tabela 11 – Interoperabilidade de ferramenta de software.....	28
Tabela 12 – Ferramentas de software a adotadas.....	34
Tabela 13 – Requisito de hardware para CityEngine.....	35
Tabela 14 – Definição de parâmetros para street.	47
Tabela 15 –Número de polígonos em cada LOD.....	52
Tabela 16 – Parâmetros de textura no CityEngine adaptado.	56
Tabela 17 – Ficheiro mtl gerados no 3DS Max e no CityEngine.	57
Tabela 18 – Canais de textura no CityEngine adaptado.	62
Tabela 19 – Descrição do ficheiro world file da imagem georreferenciada.	90

Lista de Siglas

CE – CityEngine

CGA – Computer Generated Architecture

DCC – Digital Content Creation Tools

HCD – Herança Cultura Digital

HMD – Head Mounted Display

FBX – Filmbox

OBJ – Wavefront OBJ, 3D formato para descrever geometria.

OpenGL – Open Graphics Library

2D – Bidimensional

3D – Tridimensional

3WS – CityEngine Web Scene

Capítulo 1

Introdução

O trabalho descrito nesta tese tem como temática a reconstrução virtual de espaços históricos urbanos e integra-se num projeto de investigação mais amplo que envolve a simulação de multidões neste tipo de ambiente com o intuito de recriar civilizações históricas antigas.

1.1 Motivação

A recriação digital de espaços urbanos é aplicada em diversos contextos tais como a arquitetura, o planeamento urbano, os vídeo-jogos e a herança cultural. Neste último contexto, ganha especial relevo a possibilidade de recriar digitalmente ambientes históricos total ou parcialmente desaparecidos, produzindo modelos e/ou simulações que podem ser valiosos instrumentos na investigação histórica e arqueológica, e também na divulgação cultural para o público em geral, por exemplo em museus ou através da internet.

O número de projetos realizados neste contexto é muito significativo, envolvendo na maior parte dos casos equipas multidisciplinares (arqueólogos, historiadores, informáticos e outros) e recorrendo a ferramentas especializadas. O esforço que implica uma reconstituição fidedigna e de qualidade é não negligenciável e necessita de recursos financeiros nem sempre fáceis de obter.

A motivação para este trabalho é contribuir para a definição de uma solução de baixo custo para a construção de espaços históricos urbanos que permitam a posterior inserção de personagens virtuais que recriem hábitos da época. Pretende-se identificar claramente os passos a realizar para que a solução apresentada seja facilmente reproduzida para qualquer cenário histórico e que suporte interação do utilizador em tempo real.

1.2 Objetivo

O objetivo deste trabalho é a identificação de uma sequência de etapas, que se designa por *pipeline*, que permita criar e integrar modelos digitais (por exemplo, terrenos, modelos de habitações de época ou fortificações) para obter o cenário virtual final.

Tendo em conta a preocupação de obter uma solução de baixo custo é essencial recorrer sempre que possível a ferramentas gratuitas ou com licenças académicas e que permitam fluxos de dados entre si. É, portanto, necessário identificar estas ferramentas, testando mesmo a integração de modelos tridimensionais criados por diferentes pessoas e em ferramentas distintas. Adicionalmente, considerando que o trabalho se foca em ambientes históricos, há que ter em conta que as fontes informativas podem ser variadas (escavações arqueológicas, textos, gravuras, entre outros) o que obriga a aplicar diversas estratégias para a recriação dos elementos do cenário.

Finalmente, é necessário comprovar que a solução proposta se adequa a desenvolvimentos futuros e que incluam personagens virtuais inteligentes.

1.3 Contribuições

Este trabalho começa por apresentar um levantamento sobre os conceitos relativos à herança cultural digital e sobre o estado da arte. Depois descrevem-se as ferramentas de software analisadas para a recriação digital de um ambiente histórico urbano. De seguida propõe-se um pipeline que integra estas ferramentas e mostra-se a sua aplicação num caso de estudo. Finalmente, apresenta-se o modelo integrado do ambiente virtual e os testes realizados com o modelo produzido que comprovam que a solução apresentada é válida. No decurso do trabalho foi escrito um artigo que está aceite para apresentação numa conferência internacional sobre arqueologia digital e foi submetido um outro, mais alargado, a uma revista científica da especialidade. Em ambos os artigos, é substancial a contribuição do trabalho descrito nesta tese.

A investigação levada a cabo sobre o tema herança cultural digital, em particular a reconstrução virtual de espaços urbanos históricos, deparou-se com diferentes trabalhos e projetos destinados ao resgate de património cultural. Estes trabalhos são apresentados de acordo com uma categorização definida: modelação manual, reconstrução por fotogrametria e por *laser scanning* e plataforma interativas. Para além desta

categorização, apresentam-se exemplos considerados relevantes. Na descrição das ferramentas de software potencialmente adequadas ao trabalho objetivam-se as funcionalidades específicas de cada uma e a possibilidade de interoperabilidade dos dados tridimensionais.

Como referido anteriormente, este trabalho integra-se num projeto mais abrangente de um grupo de investigadores do MAS-BioISI que se dedica à simulação de multidões usando algoritmos de Vida Artificial (um ramo da Inteligência Artificial). Estes investigadores recorrem ao Unity 3D para desenvolver os seus trabalhos. Deste modo foi obrigatório ter em conta que o modelo urbano virtual tem de ser descrito num formato suportado pelo Unity 3D. Este aspeto foi, obviamente, relevante na escolha das ferramentas a usar no *pipeline*. Um outro aspeto tido em consideração foi o terreno do modelo que se pretendia que fosse realista, tanto quanto possível próximo do original.

Assim, o *pipeline* proposto integra as seguintes ferramentas: ArcGIS para a aquisição e processamento de informação geográfica e do relevo, com a finalidade de obter um modelo virtual preciso do terreno; AutoCAD, Blender e 3DS Max para a modelação tridimensional manual de estruturas arquitetónicas e o CityEngine para modelação procedimental e integração do cenário tridimensional completo. A utilização do CityEngine permite a geração semi-automatizada de estruturas 3D através de ficheiros de regras e a possibilidade de inserção no cenário de uma vasta diversidade de modelos tridimensionais de árvores disponíveis numa biblioteca.

O caso de estudo específico ao qual foi aplicada a solução proposta é a vila de Mértola, uma vila museu localizada em Portugal, na região do Baixo Alentejo, durante o seu período de ocupação Islâmica (Séc. VII a Séc. XIII). Começou-se por obter o modelo digital do terreno recorrendo ao sítio *Open Topography* e os modelos dos edifícios foram construídos fundamentalmente com base em informação fornecida por arqueólogos do Campo Arqueológico de Mértola. Alguns destes modelos foram construídos usando modelação manual por alunos do mestrado em Informática e da licenciatura em Tecnologias de Informação do Departamento de Informática da FCUL. Outros modelos foram gerados por regras no CityEngine, plataforma na qual também foi feita a integração completa, podendo o modelo final ser visualizado no CityEngine *web scene* e plataforma Unity 3D.

Depois de exportado para o Unity 3D, procedeu-se a um primeiro teste que prova a validade do *pipeline* proposto e do modelo produzido: o ambiente virtual construído

no CityEngine foi exportado para o Unity 3D e inclui-se um avatar no cenário. Este avatar navega no cenário de acordo com a interação do utilizador em tempo real, estando ativo o mecanismo de deteção de colisões fornecido pelo motor de Física do Unity 3D. Adicionalmente, um segundo teste foi realizado por outros elementos da nossa equipa de investigação que obtiveram resposta do sistema em tempo real ao testar o cenário com grupos de personagens virtuais autónomas cujo comportamento se baseia na execução de *scripts* de Inteligência Artificial.

No decorrer do período de investigação foram escritos dois artigos científicos para os quais foi substancial a contribuição do trabalho descrito nesta tese. O primeiro foi aceite para publicação: A. P. Cláudio, M. B. Carmo, A. A. Carvalho, W. Xavier, R. F. Antunes, “*Virtual cities inhabited by autonomous characters: a pipeline for their production*”. Aceite para apresentação no 8th International Congress on Archaeology, Computer Graphics, Cultural Heritage and Innovation ‘ARQUEOLÓGICA 2.0’ in Valencia (Spain), Sept. 5 – 7, 2016.

O segundo foi submetido a uma revista científica da especialidade e encontra-se em fase de revisão, tratando-se de um artigo mais alargado e dos mesmos autores, com título: *Virtual Intelligent Characters in a Recreated Historical Urban Scene: Steps to Create the Complete Scenario*.

Acrescente-se que o trabalho decorreu entre 1 de outubro de 2015 e 30 de junho de 2016, seguindo de perto o planeamento inicial proposto, sem que tenha havido desvios significativos das tarefas realizadas dentro deste período de tempo.

1.4 Organização do documento

O documento está organizado da seguinte forma:

- Capítulo 2 – Enumera os conceitos base, as técnicas habituais de reconstrução dos ambientes urbanos digitais e o apresenta trabalho relacionado.
- Capítulo 3 – Descreve as ferramentas de software analisadas e identifica o *pipeline* a ser implementado alinhado ao caso de estudo de Mértola Virtual.
- Capítulo 4 – Detalha a implementação do *pipeline* de geração de ambiente urbano virtual.
- Capítulo 5 – Apresenta as conclusões e ideias para trabalho futuro.

Capítulo 2

Conceitos bases e trabalhos relacionados

Neste capítulo apresenta-se a definição de herança cultural digital e alguns princípios inerentes, com base em diversas fontes: a UNESCO¹, a carta de preservação da herança cultural digital de 2003, a carta de Londres² (*London Charter*) de 2009 e os princípios estabelecidos na carta de Sevilha³ (*Seville Charter*) de 2012. Estes dois últimos textos têm ambos como meta, assegurar rigor técnico e intelectual para todo o material produzido no âmbito da herança cultural digital, sendo a carta de Sevilha, mais específica relativamente à arqueologia virtual.

Descrevem-se diversas abordagens de construção de cenários virtuais, com especial foco na reconstrução digital de espaços históricos e no trabalho relacionado referem-se projetos sobre este tema.

2.1 Herança cultural digital

O conceito de Cultura surge na Grécia clássica, baseada na formação do homem grego – um homem consciente de vontade e com base na razão. Jaeger (1986), escreve: “Por mais elevadas que julguemos as realizações artísticas, religiosas e políticas dos povos anteriores, a história daquilo a que podemos com plena consciência chamar cultura só começa com os gregos”. Na Grécia clássica, a Cultura estava associada à formação do homem grego, à sua melhoria e ao seu refinamento constante. O ideal de

¹ UNESCO, United Nations Education, Scientific and Cultural Organization. Conferência Geral da Organização das Nações Unidas para a Educação, Ciência e Cultura, reunida em Paris de 17 de outubro a 21 de novembro de 1972.

² London Charter, acessado a 28 de junho de 2016, em: <http://smartheritage.com/seville-principles/london-charter>

³ Principles of Seville, acessado a 28 de junho de 2016, em: <http://smartheritage.com/seville-principles/seville-principles>

cultura para os gregos era manter e desenvolver uma cultura centrada no indivíduo. Assim, através de formação individual, forma-se uma cultura, forma-se uma nação.

A Herança Cultural pode ser considerada como a transição de uma cultura entre povos, de uma geração às seguintes. Uma cultura viva e formativa deve estar aberta para o futuro, mas ancorada no passado (Abbagnano, 2012).

No decorrer do século XX, o avanço da tecnologia, no geral, e da informática, em particular, possibilitou o surgimento de uma nova abordagem: a Herança Cultural Digital. Com o desenvolvimento do processamento computacional, torna-se possível, por exemplo, recriar modelos tridimensionais de objetos ou edifícios artísticos, com objetivo de preservar a memória do passado e realizar a transição para o presente e, possivelmente, direcioná-la para o futuro.

Estas novas possibilidades fizeram surgir iniciativas internacionais para que investigadores refletissem sobre o tema e sugerissem boas práticas no que se relaciona com o património cultural da humanidade. Este património consiste em valores de uma dada cultura e que podem possuir reconhecimento mundial. A UNESCO, organização mundial que avalia e classifica determinado património como “Património da Humanidade”, definiu Herança Cultural Digital numa convenção realizada em 1972 e do seguinte modo:

“the use of computer-based interactive technologies to record, preserve, or recreate artefacts, sites and actors of historic, artistic and cultural significance, and to deliver the results openly to a global audience in such a way as to provide formative educational experiences through electronic manipulations of time and space” (Stone, R.J. 1999).

E na sua carta de preservação da Herança Cultural Digital.⁴ (*Charter on the Preservation of Digital Heritage*) de outubro 2003 pode ler-se:

“The digital heritage consists of unique resources of human knowledge and expression. It embraces cultural, educational, scientific and administrative resources, as well as technical, legal, medical and other kinds of information created digitally, or converted into digital form from existing analogue resources. Where resources are “born digital”, there is no other format but the digital object. Digital materials include texts, databases, still and moving images, audio, graphics, software and web pages, among a wide and growing range of formats. They are frequently ephemeral, and require purposeful production, maintenance and management to be retained. Many of these resources have lasting value and significance, and therefore constitute a heritage that should be protected and preserved for current and future generations. This ever-growing

⁴ UNESCO, “Charter on the Preservation of Digital Heritage”, acedido a 16 de novembro de 2015, em: http://portal.unesco.org/en/ev.php-URL_ID=17721&URL_DO=DO_TOPIC&URL_SECTION=201.html

heritage may exist in any language, in any part of the world, and in any area of human knowledge or expression.”

A carta de Londres tem como objetivo promover a produção e disponibilização ao público em geral de material relativo à herança cultural digital; os princípios da carta de Sevilha destacam a importância de garantir a aplicabilidade da carta de Londres, com especial preocupação no domínio da arqueologia virtual. Mais detalhes sobre estes documentos podem ser encontrados no anexo H desta tese.

Ao produzir cenários de herança cultural digital, o maior desafio é desenvolver aplicações que garantam, por um lado, o resgate de património histórico da Humanidade e, por outro lado, proporcionem uma boa usabilidade por parte das pessoas envolvidas no processo.

2.2 Técnicas de reconstrução virtual

Nesta secção apresentamos técnicas de reconstrução virtual: por modelação manual, por fotogrametria e *laser scanning*, por modelação procedimental e em plataformas interativas. Em cada secção descrevemos sumariamente a técnica, apresentamos as ferramentas de software, exemplos de aplicação e apresentamos uma tabela, sempre que possível, que sumariza características particulares.

Na impossibilidade de sermos exaustivos, os trabalhos apresentados são exemplos significativos de reconstrução virtual de monumentos e/ou sítios históricos.

2.2.1 Modelação manual

Na reconstrução por modelação manual recria-se um cenário que pode ou não existir. No caso de ser um cenário histórico não existente, é necessário investigar fontes históricas e/ou arqueológicas e consultar especialistas para produzir um ambiente virtual o mais credível possível.

Neste caso, a construção dos modelos virtuais 3D é feita recorrendo a ferramentas de software de modelação e animação, por exemplo AutoCAD, Maya, 3DS Max ou Blender. Neste caso não é obrigatório que o edifício exista e há ferramentas de software gratuitas (por ex. Blender).

A Tabela 1, apresenta alguns trabalhos relacionados ao processo de recriação de herança cultural digital através de modelação manual.

Ref.	Processos	Software	Vantagens	Desvantagens
Lercari <i>et al.</i> (2013)	Criado modelo 3D de edifícios através de software 3DS Max com base em desenho CAD (gerados a partir de pesquisas históricas). Uso do motor de <i>rendering</i> Unity 3D.	3DS Max Blender Maya	Utilizações de tais softwares, permitem modelação próxima da realidade. A utilização (recriação) independe da existência do objeto a ser reconstruído.	Alguns são proprietários, exceto o <i>Blender</i> . A precisão geométrica é um desafio para os utilizadores.
Cláudio <i>et al.</i> (2013)	Modelação 3D no Blender.	Blender.	O uso de modelação zero, não exige que o objeto existe.	Dificuldade de manter precisão geométrica.
Haas <i>et al.</i> (2013)	Modelação 3D.	ArchCard	Através de fotos históricas, torna-se possível reconstruir o edifício.	Herança arquitetónica apenas. É necessário haver registos históricos.
Kennedy <i>et al.</i> (2013)	Modelação 3D.	Software Blender.	Software livre, que permite a reconstrução da Catedral.	Adquirir precisão geométrica.
Magenat-Thalman <i>et al.</i> (2006)	Modelação 3D no CAD e exportado para 3DS Max	3D Studio Max.	Utilizado a modelação 3D para projetar ambiente virtual.	Tratamento de polígonos. Alcançar precisão geométrica.
Rao <i>et al.</i> (2013)	Modelação 3D. Uso de impressora 3D para criar protótipo.		Utiliza-se do AutoCAD para criar e modelar o edifício.	Alto custo de equipamento de impressão 3D.

Tabela 1 – Processos de modelação aplicado a reconstrução por modelação raiz.

Em (Lercari *et al.* 2013) descreve-se a recriação da cidade de *Fort Ross*, localizada nos Estados Unidos da América. Atenta-se a tomada de decisão dos autores, devido ao resgate histórico do cenário e eficiência de disponibilidade em tempo real porque se pretendia obter um vídeo-jogo (tipo jogo sério). A reconstrução realizou-se com base em fontes históricas, tais como edifícios, personagens, artefactos e adereços.

Em (Cláudio *et al.*, 2013) apresentam-se modelos 3D do Observatório Astronómico da Universidade de Lisboa feitos por vários processos. Um deles recorre à

modelação manual no *Blender* e posterior exportação para X3D para torná-lo disponível pela Web e possibilitar interação do utilizador.

O trabalho de (Haas *et al.*, 2013), apresenta a recriação tridimensional da cidade de Dresden, localizada na Alemanha, na época anterior à sua destruição durante a segunda guerra mundial

Na reconstrução da Catedral de St. Andrew, na Escócia, descrita em (Kennedy *et al.*, 2013) foi utilizada modelação manual em Blender uma vez que a catedral se encontra atualmente em ruínas. Este trabalho volta a ser referido numa das secções seguintes.

Em (Magnenat-Thalmann *et al.*, 2006) é apresentada a reconstrução do teatro de *Aspendos*, na Turquia e da cidade romana antiga de *Pompeii*⁵. Os autores, utilizam modelação manual e poligonal, disponível no 3DS Max⁶.

No trabalho de (Rao *et al.*, 2013), usa-se AutoCAD para modelar Virupaksha Bazaar, em Karnataka, Índia. O grupo de monumentos pertence à lista de Património Mundial da Unesco. A reconstrução digital inclui desenhos em 2D e modelos em 3D para cada estrutura, vistas panorâmicas de diferentes pontos da estação da rua e passo a passo da rua em forma de vídeo narrativa. uma área de cerca de vinte e seis quilómetros quadrados no vale do *Tungabhadra*.

2.2.2 Fotogrametria e *laser scanning*

A modelação por fotogrametria parte da captura de imagens de um determinado objeto segundo diversos ângulos e do posterior “costura” destas imagens para produzir um modelo 3D. Na herança cultural arquitetónica o recurso à fotogrametria é muitas vezes combinado com outras técnicas, por exemplo, *laser scanning*, que captura nuvens de pontos 3D dos objectos e/ou edifícios. Estas nuvens, tratadas num software específico permitem a construção do modelo tridimensional. Os dados recolhidos por estes dois processos são reais e precisos quanto à geometria. Contudo, estas técnicas obrigam à existência física do objeto, que uma equipa técnica se desloque ao local quando o objecto de estudo estiver disponível e envolvem equipamentos e software de alto custo.

⁵ UNESCO, “Archaeological Areas of Pompei, Herculaneum and Torre Annunziata”, acessido a 20 de outubro de 2015, em: <http://whc.unesco.org/en/list/829/>

⁶ 3DS Max, “3D modelling and animation tools”, acessido a 20 de outubro de 2015, em: <http://www.autodesk.com/products/3ds-max/features/all/gallery-view>

O Tabela 2 apresenta as vantagens e desvantagens da utilização de fotogrametria para recolha de dados para modelação 3D.

Ref.	Processos	Software	Vantagens	Desvantagens
(Cláudio <i>et al.</i> , 2013)	Fotogrametria interativa e automática.	<i>Photomodels</i> (processo interativa). <i>VisualSFM</i> (Processo automático)	Modelos finais precisos e rigorosos geometricamente.	Equipamento de auto custo. Necessário a existência física do objeto.
(Micoli <i>et al.</i> , 2013)	Fotogrametria	TrueView	Repositório com dados de nuvens de pontos para análise da equipa.	Aquisição de equipamento de auto custo. Deslocamento.
(Magnenat-Thalman <i>et al.</i> , 2013)	Cria uma biblioteca de fotografia de alta resolução.	3D Studio Max	O uso de uma biblioteca de fotografias permite trabalhar textura no objeto 3D.	Deslocar ao local. Não aplicável quando objeto não estiver disponível.
(Rao <i>et al.</i> , 2013).	Fotogrametria	AutoDesk ImageModeler	Modelagem 3D precisa.	Necessita de equipamentos e de software. O objeto deve existir para desenvolver sua reconstrução.

Tabela 2 – Processos de modelação aplicado a reconstrução por fotogrametria.

Um exemplo é o trabalho descrito em (Micoli et al, 2003) em que a criação de modelo virtual está fortemente focada por captura de dados através de *laser scanning*, para gerar nuvens de pontos. No entanto, utiliza-se fotogrametria para criar uma base de dados digital de imagens para apoiar o processo de texturização. Algumas imagens para criar nuvens de pontos 3D coloridas foram capturadas através da técnica SFM (estrutura em movimento), que pode ser incluído na técnica de fotogrametria terrestre. Equipamento de alto custo.

A fotogrametria permite duas abordagens de implantação/utilização: a fotogrametria interativa e fotogrametria automática. No caso de estudo de (Cláudio *et al.*, 2013), certifica-se a utilização de processos de fotogrametria interativa e automática para recolha de dados do edifício, fazendo uso do software *PhotoModeler* e fotogrametria automática, que utiliza o software *VisualsFM*. As grandes características destas abordagens consistem em modelos finais geometricamente precisos, de acordo com estrutura física do objeto.

Em (Rao *et al.*, 2013) é descrita a reconstrução através de fotogrametria na reconstrução dos monumentos de *Hampi*⁷. Acrescentou-se ao modelo, todos os recursos desenvolvidos (fotografia, fotogrametria, modelo 2D, modelo 3D e vídeo) e tudo o material se encontra disponível numa plataforma Web que permite a interação de utilizadores.

Troiano *et al.* (2014) utilizaram a técnica de *laser scanning* para fazer o levantamento do castelo de Pietrabuona, na Toscana, Itália. Este projecto recorre a um modelo de arquitetura cliente/servidor e ao Unity 3D para disponibilizar uma interface interativa ao utilizador na Web. Estes autores adicionam a técnica de *laser scanning* com recurso de fotogrametria para criar uma base digital de imagens (ver Figura 4) para usar no processo de texturização.

O complexo monumental de túmulos do *Hypogeum of the Gorgons*, em Perúgia, Itália, foi construído usando combinações de diversos elementos e de técnicas que permitiram a criação de um ambiente 3D para interação por parte do utilizador. Recorreu-se a fotogrametria para documentar as condições dos espaços e o mapeamento de textura nos modelos 3D. O mapeamento de textura do modelo tridimensional foi adquirido por varredura a laser (Gabellone *et al.*, 2013). Foi usado o software *VirTools*, que possui bibliotecas gráficas reutilizáveis e que possibilita desenvolver um ambiente interativo. No entanto, o software *VirTools* foi descontinuado e substituído no mercado pelo *3DVIA Studio Pro*⁸.

Na Tabela 3, apresenta comparativos entre os trabalhos focados na técnica de *laser scanning*.

Ref.	Processo	Software	Vantagens	Desvantagens
Troiano <i>et al.</i> (2014)	Utilizado <i>laser scanning</i> para captura de imagens do castelo. Imagens armazenadas em base de dados.	SIUR – <i>Urban Information System</i> . Motor de renderização do 3D Unity, utilizado no projeto.	Permite armazenar imagens em bases de dado através da captura por <i>laser scanning</i> . Interação pela web, através de carregamento	Dificuldade de gestão de base de dados. Pois, as imagens dever ser trabalhadas quanto a qualidade e tamanho do arquivo.

⁷ UNESCO, “Group of Monuments at Hampi”, acessido a 12 de outubro de 2015, em: <http://whc.unesco.org/en/list/241/>.

⁸ Dassault Systemes, acessido a 28 de outubro de 2015, em: <http://www.2.3ds.com/products-services/3dvia/3dvia-studio/welcome/>

			dos links para visualizar ambiente 3D.	
Micoli <i>et al.</i> (2013)	Utilizado <i>laser scanning</i> para recolha de dados.	Desenvolvido uma plataforma web - ISEE, que permite navegabilidade e interação do utilizador.	Permite armazenar dados de recolha em base de dados. Utiliza motor de renderização Unity 3D para disponibilizar modelo pela web.	Utilização de equipamento de alto custo. Análise e validação da historicidade de mudança do edifício.
Gabellone <i>et al.</i> (2013)	Utilizado informação de dados fotográficos manipulada em conjunto por <i>laser scanning</i> .	3DVIA Virtools	A ferramenta VirTools permite disponibilizar reconstrução para Web.	Utilização de equipamento para varrimento e acesso ao objeto a reconstruir. Utilização de diversos software para modelação 3D.

Tabela 3 – Trabalho relacionado ao processo de *laser scanning*.

2.2.3 Modelação procedimental

A modelação procedimental integra técnicas para gerar modelos 3D e texturas automaticamente recorrendo a regras e existem várias abordagens.

O trabalho referido em Muller *et al.* (2006) define uma gramática de forma, CGA⁹, recorre ao CityEngine e pode-se constatar a importância do módulo utilizado para resgatar a cidade antiga romana de *Pompeii*.

O trabalho de (Rodrigues, 2008) apresenta um outro exemplo, que faz uso da gramática formal na reconstrução do sítio de Conímbriga. Usa conceitos de arquitetura Greco-Romana de Marcus *Vitruvius*¹⁰, que estabelece um padrão de desenvolvimento de

⁹ CityEngine Help, CGA – Computer Generated Architecture, acedido a 08 de novembro de 2015, em: <http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/cgareference/cgaindex.html>

¹⁰ Arquiteto romano que viveu no século I a.C. Autor da obra *The Architecture*. Acedido a 07 de outubro de 2015, em: http://www.forgottenbooks.com/readbook/The_Architecture_of_Marcus_Vitruvius_Pollio_v1_1000069524#267

acordo com a época. O foco está em criar modelos usando técnica escrita de regras gramaticais para automatizar a reconstrução seguindo evidências fidedignas da época romana. Define-se inicialmente uma estrutura padrão de casas romanas, com abordagem de regras gramaticais. As regras possuem duas vertentes: a) regras de construção, de responsabilidade do *L-System*¹¹; b) regras de reconstrução, de responsabilidade do utilizador (Rodrigues et al., 2014).

A Tabela 4 apresenta uma diversidade de trabalhos com enquadramento de modelação procedimental e com foco específico nas áreas de criar terreno, vegetação, rio, rede de estradas, *layout* de cidade, construção de edifício.

Ref.	Vegetação	Estradas	Cidade	Construção	Interior
Besuievsky, 2013)			X	X	
Parish <i>et al.</i> (2001)		X	X		
Lipp <i>et al.</i> (2008)				X	
Rau-Chaplin <i>et al.</i> (1996)				X	X
Longay <i>et al.</i> (2012)	X				
Rodrigues <i>et al.</i> (2008 e 2014)			X	X	
Muller <i>et al.</i> (2006)		X	X	X	
Patow et at. (2012)				X	
Dylla <i>et al.</i> (2010)			X	X	
Stouffs <i>et al.</i> (2015),				X	X

Tabela 4 – Categorização por especificidade de modelling procedural.

O software *CityEngine* possui abordagem baseada em *L-System* (extensão para produção de regras) para construção de cidades. A característica principal, está na sua arquitetura de objetivos globais e restrições locais, com criação por padrões de cidade e densidade populacional e limites de terra, água, elevação, cruzamento de ruas, entre outros (Parish e Muller, 2001).

Em (Lipp, 2008) apresenta-se um editor visual para gerar regras de gramática que permite fácil interação do utilizador. Permite a definição de regra base e flexibilidade na modificação.

O trabalho de (Rau-Chaplin, 1996) aplica uma gramática formal para gerar a planta baixa de uma casa, contendo unidades básicas no seu interior.

¹¹ L-System ou sistemas de *Lindenmayer* (em homenagem a seu criador) consistem gramáticas formais e foram introduzidos em 1968 pelo húngaro teórico biólogo e botânico da Universidade de Utrecht, Aristid Lindenmayer. Resume-se em sistema de reescrita de símbolos terminais e não terminais.

O caso de estudo de (Logay *et al.*, 2012) usa a modelação procedimental para contruir diversidade de árvores, através de uma interface gráfica interativa. Trata-se de aplicação *TreeSketch*¹², que permite a simulação e criação de uma variedade de árvores.

O trabalho de (Muller *et al.*, 2006) apresenta uma plataforma de reconstrução da cidade antiga de *Pompeii*, utilizando modelação procedimental. Usa *CGA-Shape* (uma extensão de *L-System*) integrada no *CityEngine*.

O trabalho de (Patow *et al.*, 2012) apresenta uma interface amigável ao utilizador e que possibilita a criação de edifício pela edição de um grafo. A aplicação é uma extensão de funcionalidade da ferramenta de animação 3D *Houdini*¹³, uma ferramenta para criação de modelação de edifícios, um protótipo totalmente funcional para a geração de edifícios baseados em grafos que utiliza *python scripts* incorporados.

No caso de estudo de (Dylla *et al.*, 2010) utiliza-se o software *CityEngine* para definir um código de regras para a reconstrução de cidade antiga de Roma.

O trabalho de (Stouffs *et al.*, 2015) usa a linguagem XML para descrever as regras gramaticais que lhe permitem obter um modelo 3D de Mesquitas no período Otomano.

Na reconstrução por regras gramaticais adota uma variedade de soluções para atender à necessidade de diversas especificidades dos modelos. A integração de tais soluções torna a criação de ambientes virtuais (mundo virtual) mais eficiente.

Na Tabela 5 regista-se o contributo de da aplicação de regras no *CityEngine* em trabalhos no contexto da reconstrução de herança cultural digital:

Autor	Contributo	Módulo/Método	Âmbito
Parish <i>et al.</i> (2001)	Definição de regra para construção de cidades virtuais.	<i>CityEngine System</i> – Sequência de módulos específicos para construção de cidade virtual.	Construção de cidades inteiras virtual.
Wonka <i>et al.</i> (2003)	Definição de regras para reconstrução de fachadas de edifícios.	Framework para modelação de arquitetura de edifício individual.	Construção de edifícios de complexidade geométricas individuais.
Muller <i>et al.</i> (2006)	Definição de regras gramaticais para reconstrução de cidades em massa.	<i>CGA Shape</i> integrado ao <i>CityEngine</i> .	Construção de modelos tridimensional em larga escala.

Tabela 5 – Principais contributos para o CityEngine.

¹² *TreeSketch*, acedido a 29 de outubro de 2015, em: <http://algorithmicbotany.org/TreeSketch/>

¹³ *Procedural Tools for Animation, VFX & Game*. Acedido a 20 de outubro de 2015, em: https://www.sidefx.com/index.php?option=com_content&task=blogcategory&id=117&Itemid=374

A seguir, apresenta-se uma análise prática de três projetos relacionados a reconstrução de herança cultural digital apresentado acima.

2.2.4 Plataforma interativas

Esta secção apresenta soluções que integram diversas técnicas utilizadas para reconstruir herança cultural digital e que recorrem a ferramentas com motor de *rendering* integrado. Acrescenta-se a este tópico, alguns trabalhos que contemplam realidade aumentada.

A plataforma *OpenSim*¹⁴, de código fonte aberto, permite a customização e a criação de um ambiente virtual que suporte a reconstrução de herança cultural digital. Em (Kennedy *et al.*, 2003), constata-se que *OpenSim* oferece toda estrutura para projetar ambiente virtual aberto e que fica disponível para o utilizador interagir através de avatares¹⁵. O *SecondLife* é um outro software nesta linha, mas de domínio proprietário. Ambos permitem criar ambientes virtuais, porém, *OpenSim* é livre e permite maior abordagem personalizável na criação de um ambiente virtual. Estas plataformas obrigam a que exista um servidor com capacidade de suportar interação com o ambiente virtual em tempo real.

O projeto de reconstrução da Catedral de *St. Andrews*¹⁶, que se encontra em ruínas, ganhou vida em ambiente virtual, utilizando-se dos recursos do *OpenSim* e usando um avatar para representar a interação com ambiente virtual (Kennedy *et al.*, 2013). A Tabela 6, apresenta uma comparação entre os ambientes virtuais *OpenSim* e *SecondLife*. Destaque-se que nalguns critérios são mais vantajosos na plataforma livre *OpenSim*, que permite autonomia de configuração e personalização do ambiente.

No trabalho de (Fourts, 2010) usa-se a plataforma *SecondLife* para reconstruir um complexo do *Al-Andalus*, localizado em Granada, Espanha. O projeto tem como foco restabelecer a convivência de diversas culturas numa plataforma virtual do *SecondLife*, com o objetivo de compreender a cultura islâmica através do mundo virtual – criação da

¹⁴ OpenSimulador: OpenSim is an open source multi-platform, Multi-user 3D. Acedido a 12 de outubro de 2015, em: http://opensimulator.org/wiki/Main_Page.

¹⁵ Representa interação real (atual) do modo de vida social. Isto é, o utilizador identifica-se com a escolha de um personagem, e pode interagir dentro do ambiente virtual aberto, neste caso, conhecer os detalhes da reconstrução de *St. Andrews Cathedral*. Acedido em 09 de outubro de 2015, em: <http://blogs.cs.st-andrews.ac.uk/openvirtualworlds/reconstructions/st-andrews-cathedral/>

¹⁶ St. Andrews Cathedral 1318 – Open Virtual Worlds. Acedido a 12 de outubro de 2015, em: <http://www.openvirtualworlds.org/reconstructions/st-andrews-cathedral/>

cidade história de *Alhambra*¹⁷ no período de ocupação Islâmica em Espanha entre os séculos XIII e XIV.

Tópico	OpenSim	Secondlife	Análise
Custo	Free	Proprietário	OpenSim não oferece custo. Necessário somente haver servidor (hardware).
Linguagem de programação	C# e LSL	Apenas LSL (<i>Linden Scripting Language</i>)	Possibilita realizar extensões com versão livre. Através da linguagem de programação C# possibilita acrescentar novas funcionalidades.
Firewall	Liberdade para configurar as portas necessárias. O servidor pode estar na intranet da organização.	Necessita de permissão de portas específicas de acesso.	O servidor pode fazer parte dos ativos da organização.
Customização	Permite personalizar aplicações de mundo virtual aberto.	Há limitações quanto a personalização.	A plataforma <i>OpenSim</i> , permite personalizar de acordo com a necessidade de um determinado cenário virtual.
Humano virtual ou Avatar	Avatar Pode usar qualquer nome.	Avatar (deve usar sobrenome do laboratório)	As duas plataformas utilizam avatar.

Tabela 6 – Comparativo entre as plataformas OpenSim e Secondlife.

2.3 Trabalhos relacionados

Nesta secção apresentam-se alguns trabalhos agrupados do seguinte modo: exemplos que recorrem à realidade virtual, alguns deles com humanos virtuais, exemplos com realidade aumentada e, um último grupo que corresponde a trabalhos mais próximos do desenvolvido nesta tese.

¹⁷ Aramco World, “A virtual walking tour: the Alhambra”, acessado a 15 de outubro de 2015, em: <http://www.aramcoworld.com/issue/200604/alhambra/default.htm>.

2.3.1 Exemplos com realidade virtual

A presente secção enquadra trabalhos de herança cultural digital que usam a realidade virtual. Tabela 7, comparam-se as principais características de um conjunto de trabalhos de reconstrução de herança cultural digital em ambientes virtuais.

Ref.	Cenário de reconstrução	Interação com Ambiente Virtual	Aplicação/Software	Humano Virtual
Silva <i>et al.</i> (2010)	Casa dos Esqueletos – Conimbriga	Sim	VRML	Não
Magenat-Thalman <i>et al.</i> (2006)	Teatro de Aspendos e Pompeii	Sim	Framework VHD++ MR	Sim
Lercari <i>et al.</i> (2013)	Fort Ross - paisagem histórica	Sim	3D Studio Max e Motor de jogos do Unity 3D - Fort Ross Armazém Virtual jogo sério desenvolvido em Unity 3D.	Sim. Personagens virtuais.
Troiano <i>et al.</i> (2014)	Castelo de Pietrabuona	Sim	Software com arquitetura cliente/servidor e motor de renderização do Unity 3D.	Não.
Rao <i>et al.</i> (2013)	Conjunto de monumentos de Hampi	Sim	Ferramenta baseada na web – arquitetura cliente/servidor.	Não.
Kennedy <i>et al.</i> (2013)	Reconstrução da Catedral de St. Andrew	Sim	Plataforma virtual multiplataforma de código fonte aberto	Sim. Avatar.
Gabellone <i>et al.</i> (2013)	Complexo monumental de túmulos	Sim.	Plataforma <i>VirTools</i>	Sim. Personagem virtual estático.
Dylla <i>et al.</i> (2013).	Reconstrução virtual de Roma antiga.	Sim.	CityEngine e Mental Ray para renderização do ambiente virtual.	Não.

Tabela 7 – Relação de trabalhos inerente a realidade virtual.

O projeto de reconstrução da casa de esqueletos, no sítio arqueológico de Conimbriga, recorre à tecnologia VRML, está disponível na Web e é interativo¹⁸ (Silva *et al.*, 2004). Os autores tiveram em conta três fatores principais: realismo, tamanho do ficheiro e desempenho.

A reconstrução do teatro de *Aspendos*, na Turquia e a cidade romana antiga de *Pompeii* são abordados em (Magenat-Thalmann *et al.*, 2006). A construção do modelo tridimensional realiza-se através da técnica de modelação poligonal e para definir o comportamento do humano virtual definido no cenário utiliza-se um *python script*.

O trabalho elaborado por (Lercari *et al.* 2013) caracteriza-se pela utilização do motor de jogos do Unity 3D *game engine*. Proporciona um ambiente virtual para interação do utilizador. O autor recorre a elementos gratuitos ou de baixo custo: modelos tridimensionais, personagens manipulados, *plug-ins*, animações, extensões de desenvolvimento. Estes podem ser obtidos gratuitamente ou adquiridos *on-line* por meio do *Asset Store Unity* a um preço conveniente e, em seguida, importados diretamente na cena tridimensional do ambiente Unity 3D.

A arquitetura cliente/servidor definida em (Troiano *et al.* 2014), possibilita o armazenamento de dados capturado em banco de dados, e que são disponibilizados em uma interface web.

No trabalho elaborado por (Rao *et al.*, 2013) usa um ambiente Web – arquitetura cliente/servidor que possibilita o armazenamento de dados identificados pelo ID, o que permite disponibilidade por uma interface ao utilizador realizar interação com o conjunto de monumentos de *Hampi*.

A reconstrução da Catedral de St. Andrew (cenário do séc. XIV) possibilita a navegação no interior do edifício e no exterior através de avatares, um deles em 1ª pessoa (Kennedy *et al.*, 2013). A plataforma *OpenSim* permite o registro de diversos utilizadores em simultâneo.

Em (Gabbelone *et al.*, 2013) estabelece-se um ambiente virtual que permite a navegabilidade do utilizador e a exploração do cenário reconstruído. O sistema de navegação utiliza-se de um interface natural e comandos executados com gestos simples, utilizando os recursos do *VirTools*. Ao utilizar esta tecnologia – o software *VirTools* possui compatibilidade com a biblioteca *OpenGL* e caracteriza-se por ser um

¹⁸ A casa dos Esqueletos de Conimbriga, acedido a 29 de outubro de 2015, em: <http://www.conimbriga.pt/portugues/ruinas09.html>

motor de tempo real – os visitantes são capazes de seguir um itinerário integrado de conhecimento baseado em conteúdo heterogêneo, incluindo dados históricos e arqueológicos e os resultados das análises arqueológicas. Destaca-se a característica de abordagem ativa (o utilizador interage com aplicação através de uma interface amigável) e passiva (vídeo com narração do cenário apresentado ao utilizador).

O projeto de *Rome Reborn*, caracteriza-se por desenvolver a reconstrução da cidade antiga de Roma (Dylla *et al.*, 2010) numa plataforma virtual que permite o utilizador interagir através de uma interface e explorar a cidade antiga de Roma.

Os trabalhos apresentados na Tabela 8, inclui-se como critério a utilização de humano virtual ou avatar na cena de realidade virtual. A utilização de humano virtual em ambiente de realidade virtual, possui como objetivo compreender o uso da estrutura cultural de uma determinada época, projetar uma simulação e/ou reconstrução de herança cultural digital mais realista.

As aplicações de herança cultural virtual que utilizam humano virtual tiveram início com o projeto *Terracotta Soldiers*. Sobre humanos virtuais, Magnenat-Thamann (2004, p. 38) escreve:

One of the first example of the use of Virtual Humans in Virtual Heritage was the simulation of the Terracotta soldiers [http: TerraCotta] by MIRALab, University of Geneva. Excavation of the grave complex of the Ch'in emperor Shi Huang Ti in Xian in the 1970s revealed a vast field of lifesize terracotta statues depicting soldiers, servants, and horses, estimated to total 6,000 pieces.

Os humanos virtuais são utilizados para tornar um ambiente virtual cada vez mais próximo da realidade. Para tornar impercetível o movimento de humano virtual em comparação a humano na vida real, as pesquisas avançam para proporcionar o surgimento de novas tecnologias, como por exemplo *Motion Capture*.

Os projetos de (Magnenat-Thamann *et al.* 2006 e Papagiannakis *et al.* 2002) são bem-sucedidos na transposição de movimentos capturados em ambiente real por seres humanos para o mundo virtual. Em ambos os casos de estudo (*Pompeii* e *Aspendos*) é realizada captura de movimento com presença de atores profissionais através da técnica/processo de *Optical Motion Capture*.

A Tabela 8 categoriza detalhes inerentes à utilização de humanos virtuais em trabalhos com foco de reconstrução de herança cultural virtual.

Ref.	MoCAP	Text to Speech	Interação com Humano Virtual	Recurso técnicos
Magenat-Thalmann <i>et al.</i> (2006)	Sim	Não	Sim	Python Script para definir comportamento do humano virtual.
Lercari <i>et al.</i> (2013)	Não	Não	Sim	Personagens virtuais presente no cenário – uso do Autodesk Maya.
Kennedy <i>et al.</i> (2013)	Não	Não	Sim	Personagens virtuais – Avatar (os script em C# são incluídos para determinar comportamento).
Gabellone <i>et al.</i> (2013)	Não	Não	Não	Personagens virtual estático.

Tabela 8 – Características específica em Humano virtual.

2.3.2 Exemplos com realidade aumentada

A realidade aumentada (RA) adiciona novos recursos computacionais para visualizar informações digitais que são sobrepostas às imagens reais, capturadas por uma câmara e vistas diretamente através dos olhos (abordagem óptica) ou com tecnologias de vídeos (abordagem vídeo).

A RA é utilizada em diversas aplicações para contemplar uma variedade de áreas (jogos, etc) e também em aplicações de herança cultural digital – por exemplo, dentro de um museu ou um sítio arqueológico. O trabalho de (Carmo & Cláudio, 2013) enumera diversos trabalhos neste âmbito, ao ar livre em sítios arqueológicos ou templos antigos, ou em espaços fechados, por exemplo, dentro de um Museu. Pode concluir-se que há limitação para ambientes externos no uso de dispositivos tecnológicos agregado à realidade aumentada (caso da movimentação livre do utilizador e a dificuldade de rastrear posicionamento - *tracking*).

Ref.	Reconstrução	Visualização	Abordagem HMD
Barazzetti <i>et al.</i> (2015)	Castelo de Masegra	Externa	HMD – Vídeo
Gleue e Dähne (2002)	Templo de Here, Grécia Antiga.	Externa	HMD – Ótica.
Vlahakis <i>et al.</i> (2012)	Templo de Her, Grécia Antiga	Externa	HMD – Ótica.

Tabela 9 – Herança cultural com realidade aumentada.

A Tabela 9 compara trabalhos de reconstrução de herança cultural que recorrem à RA (por abordagem vídeo ou óptica).

Barazzetti *et al.* (2015) desenvolvem dois cenários: a) visita virtual – aplicação iVisit3D que permite ao utilizador visualizar objeto em 3D – e realidade aumentada do Castelo de Masegra - aplicação RA.

A realidade aumentada aplicada no trabalho de (Gleue & Dähne, 2002) consiste em uma visualização externa do templo de Hera, na Grécia antiga. Utiliza-se um HMD (*Head Mounted Display*) óptico para representar o cenário de realidade aumentada. O trabalho faz parte do projeto *Archeoguide* – um projeto de realidade aumentada do templo de Hera e que além de possibilitar visualização de realidade aumentada do edifício, simula a vida antiga na Grécia, caso de avatares para representar realidade do passado – jogos no estádio antigo de *Olympia* (Vlahakis *et al.*, 2012).

Os vários autores destacam o desconforto do utilizador ao usar dispositivos de RA, como HMD.

2.3.3 Exemplos a destacar

Nesta secção damos alguns detalhes adicionais sobre 3 dos trabalhos referidos nas secções anteriores porque estes se assemelham de algum modo ao caso que queremos tratar.

2.3.3.1 A catedral de St. Andrews

O ambiente virtual denominado *Open Virtual World* desenvolvido pela *University of St. Andrews*, é uma plataforma aberta baseada em *OpenSim*. É multiplataforma e multiutilizador – permite realizar personalização/customização em diversos sistemas operativos, por exemplo Linux e Windows.

O projeto da Catedral de *St. Andrews* resgata o passado da catedral do séc. XIV e disponibiliza-o numa plataforma virtual. O utilizador necessita de realizar registo para em seguida descarregar a plataforma cliente e realizar instalação do aplicativo Apollo¹⁹.

Após a instalação, o utilizador terá permissão para aceder à plataforma virtual e poder explorar a Catedral num cenário reconstituído do ano de 1318. Um avatar implementado no ambiente virtual, permite explorar o ambiente, realizar interação com

¹⁹ Open Virtual Worlds, “Getting Start”, acedido a 03 de novembro de 2015, em: <http://blogs.cs.st-andrews.ac.uk/openvirtualworlds/create-account/download-viewer/>

outros utilizadores em tempo real, que estão simultâneamente presentes no ambiente virtual. Na Figura 1, apresenta a interface de autenticação para explorar a Catedral através do *Open Virtual Worlds*.



Figura 1 – Interface de autenticação do *Open Virtual Worlds*.

A Tabela 10 apresenta características técnicas do simulador *OpenSim*, desenvolvido na catedral de St. Andrews.

Componentes	Linguagem de Programação	Principais abordagens
Cliente Simulação Servidor web	OpenSim é escrito em C# e funciona sobre o .NET (Dot Net) Framework.	<ul style="list-style-type: none"> - Arquitetura cliente/servidor - Multiplataforma - Código fonte aberto - Multiutilizador - Extensão de funcionalidade

Tabela 10 – Características técnica do *OpenSim*.

Entre as componentes presente no *OpenSim*, destaca-se a componente de simulação, que utiliza a biblioteca *libOpenMetaverse* – esta biblioteca é uma coleção de .NET (Dot Net) escrita em C# para interação e simulação com mundo virtual 3D. As componentes identificadas na Tabela 10 são apresentados no projeto *LibOpenMetaverse*²⁰, tendo como objetivo criar ambiente virtual 3D em uma arquitetura cliente/servidor, desenvolvido em C# e .NET (Dot Net).

²⁰ Libopenmetaverse, “Libomv Developer Wiki”, acessido a 20 de novembro de 2015, em: http://lib.openmetaverse.co/wiki/Main_Page

2.3.3.2 Rome Reborn

O projecto Rome Reborn usa o CityEngine cuja principal característica é a extensibilidade de novas funcionalidades por regras gramaticais (extensão de funcionalidade do modelo *L-system*) e certifica a reconstrução das cidades de Nova York, Tóquio, Paris e Londres através de dados recolhidos e (Parish e Muller 2001).

O software CityEngine é proprietário com distribuição pela ESRI e possui uma versão *free trial* de trinta dias.

A reconstrução virtual da Roma antiga, denominada de “*Rome Reborn*”, versão 2.0 foi desenvolvida por regras gramaticais no *CityEngine*. As regras são estabelecidas e pré-definidas com base em consulta de especialistas (arqueológicos) para recolha de dados precisos e credíveis. No trabalho de (Dylla *et al.*, 2010) refere-se a reconstrução virtual, de quatorze regiões do sítio antigo de Roma. Além disso, o ambiente virtual possui disponibilidade pela Web, o que permite o utilizador aceder e interagir com a reconstrução da cidade antiga de Roma. A arquitetura para disponibilizar interface utilizador, foi desenvolvida com base no *Reality Server FLEX*.

2.3.3.3 Castelo de Pietrabuona

Este trabalho recorre ao motor de jogos do Unity 3D para disponibilizar visualização de três dimensões do resgate histórico do castelo de *Pietrabuona* (Merlo *et al.*, 2012).

Os autores usam o software SIUR 3D *URban Information Systems* que recorre a dados de imagens capturados por *laser scanning* e os carregam em bases de dados devidamente identificado (ID) para cada imagem. Através deste armazenamento, combina as imagens para gerar visualização utilizando-se da capacidade do motor de renderização do Unity 3D. Outra técnica abordada, que resulta em qualidade, consiste em dividir em bloco para construir (praças, rua, fachadas, etc.).

A conexão é realizada pela interface utilizador web para aceder sistema *Urban Information System 3D* para manipulação de objetos tridimensional.

Considera-se a técnica de parametrização de polígonos e a definição de limites para realizar a transição para o motor *Game Engine*. A definição consiste em estabelecer um limite de quatro milhões de polígono para cada objeto do castelo.

O modelo disponibilizado na Web permite exploração do Castelo *Pietrabuona* interativa através de *link* de navegabilidade. A Figura 2 representa uma captura de

visualização da frente do Castelo, identificado como número dois (ID da imagem armazenado em banco de dados) da aplicação.



Figura 2 – Interação do utilizador pela User Interface.

2.4 Conclusão

Neste capítulo apresentaram-se conceitos base e trabalhos relacionados. Através da leitura dos artigos pesquisados percebemos que para resgatar um determinado sítio arqueológico em ruínas ou totalmente desaparecido é preciso investigar o seu passado histórico e identificar registos para elaboração de modelos tridimensionais credíveis. Assim, existe neste desafio um carácter de interdisciplinaridade entre diversas áreas do conhecimento, porque para além de informáticos são necessários, por exemplo, arqueólogos e historiadores- profissionais extremamente importantes para identificar realidades no passado, e assim possibilitar a transição desta informação para uma plataforma virtual onde será disponibilizada para gerações presentes e futuras.

As recomendações das cartas de Londres e Sevilha reforçam a necessidade de rigor técnico e intelectual no desenvolvimento dos projetos de herança cultura digital, uma preocupação que tivemos em conta no nosso trabalho.

Como se disse anteriormente, não é possível referir todos os trabalhos nesta temática. Outros exemplos de reconstrução virtual de costumes e cidades antigas estão referidos no artigo (Carmo e Cláudio, 2013).

Capítulo 3

Análise de ferramentas e caso de estudo

Neste capítulo descreve-se o trabalho realizado no que concerne ao estudo de ferramentas de software e introduz-se o caso de estudo para o cenário de Mértola Virtual.

3.1 Ferramentas de software

A presente secção possui como objetivo catalogar as ferramentas de software quanto à sua utilização no âmbito da herança cultural digital. As ferramentas foram categorizadas em três tópicos: modelação tridimensional, automatização de reconstrução e motor de renderização.

Estudaram-se ferramentas de domínio publico (Blender), ou que têm um subconjunto de funcionalidades gratuitas (Unity 3D) ou para as quais existe uma licença académica (AutoCAD, 3DS Max e CityEngine). Descrevem-se em seguida cada uma dessas ferramentas:

- O AutoCAD²¹ caracteriza-se por ser uma ferramenta de modelação e desenho dimensional e tridimensional. Possui como característica o desenho de modelos de raiz, tendo a capacidade de exportar os modelos para diversos formatos. Entre estes formatos destaca-se o formato DXF (*Drawing Exchange Format*) que permite compatibilidade com 3DS Max e CityEngine.
- O Blender²² é um software com capacidade de modelação tridimensional e que permite importação e exportação de diversos formatos. Entre os principais formatos para buscar interoperabilidade com outras ferramentas, certifica-se o

²¹ Autodesk, “AutoCAD”, acedido em 12 de julho de 2016, em: <http://www.autodesk.com/education/free-software/autocad>

²² Blender, “Blender 2.77a”, acedido em 12 de julho de 2016, em: <https://www.blender.org/>

formato OBJ e FBX. O Blender permite a escrita de scripts em *Python* para definir a automatização da criação de cenas.

- 3DS Max²³ caracteriza-se por ser um software de modelação tridimensional e com recursos mais robustos que os mencionados anteriormente. Por exemplo, é versátil na aplicação de texturas e pode realizar animações e renderização de imagens. A sua versão mais recente (2016) possui melhoramentos no módulo de população e animação de humanos virtuais, acrescentando ao módulo a simulação de multidões. Possui capacidade de importar e exportar para diversos formatos. O formato OBJ permite compatibilidade com outras ferramentas como, por exemplo, o CityEngine.
- O CityEngine²⁴ consiste em uma plataforma de software para desenvolver a construção de ambientes urbanos virtuais – cidades virtuais. A sua principal característica reside em a partir do modelo 2D (*shape*) criar um modelo 3D através de *procedure rule CGA*. O recurso do ambiente de desenvolvimento CityEngine, designado por *Computer Generated Architecture (CGA)* permite aplicar *shape grammar rule files*. Além disso, tem um editor de *Python Script* integrado no ambiente CityEngine, que permite a definição de programação de *script*. Possui uma excelente capacidade de interoperabilidade de ferramentas de modelação tridimensional e animação para receber ficheiros em diversos formatos. Ademais, possui capacidade de exportar para diferentes formatos direccionados para motores de renderização, como, por exemplo, o *unity game engine* pelo formato FBX e o CityEngine *Web Scene* pelo formato 3WS. Acrescenta-se que tem compatibilidade com a plataforma ArcGIS, no que diz respeito ao processamento de informação geográfica necessária para a construção do modelo digital do terreno e respetiva textura com uma imagem georreferenciada.
- Unity 3D²⁵ é um motor de jogos, que possui como principal característica ser um ambiente de desenvolvimento de jogos multiplataforma, o que permite exportar um projeto para diversas arquiteturas computacionais. É possível

²³ Autodesk, 3DS MAX, acessido a 02 de novembro de 2015, em: <http://www.autodesk.com/products/3ds-max/features/all/gallery-view>.

²⁴ ESRI, “Esri CityEngine”, acessido a 12 de julho de 2016, em: <http://www.esri.com/software/cityengine>.

²⁵ Unity, “Unity Personal”, acessido a 12 de julho de 2016, em: <https://store.unity.com/download?ref=personal>.

definir o comportamento dos objetos da cena através de *script C#* - ambiente de programação nativa integrada na plataforma Unity 3D, entre outras particularidades, torna-se possível criar efeitos gráficos, controle do comportamento físico de objeto e de personagens através do motor de física. Admite a importação de dados tridimensionais desenvolvidos em diversas ferramentas.

Analisam-se em seguida as diferentes ferramentas relativamente à interoperabilidade de dados 3D, à modelação tridimensional, à automatização de reconstrução 3D e à compatibilidade com motores de *rendering*.

3.1.1 Interoperabilidade de dados 3D

A interoperabilidade de dados é a capacidade de os sistemas trocarem informações sem perda de dados tridimensionais. Quanto à interoperabilidade, (Bass, 2003) descreve: “a interoperabilidade é sobre a capacidade de dois ou mais sistemas poderem trocar informações”. Observa-se que o autor referido alerta que a interoperabilidade inclui não somente ter a capacidade de trocar dados –interoperabilidade sintática, mas também a capacidade de interpretar corretamente os dados que estão sendo trocados – contemplar a interoperabilidade semântica.

A interoperabilidade busca proporcionar compatibilidade entre as diferentes aplicações com a transferência de dados mantendo a sua integridade. Tendo em vista, a diversidade de aplicações utilizadas na reconstrução de herança cultural digital, para garantir a interoperabilidade entre elas, cada ferramenta contempla diversos formatos de exportação e importação.

O formato FBX (FilmBox) foi desenvolvido pela companhia Kaydara, fundada em 1993, e integrou-se na Autodesk, com denominação de Autodesk FBX, devido à aquisição da Kaydara pela Autodesk (Fläckman & Roos, 2006). O principal objetivo consiste em prover interoperabilidade de dados tridimensional entre a Autodesk e os demais softwares de concepção de conteúdo digitais com formato 3D. Pode consultar-se a comparação deste formato com outros no trabalho de McHenry et al. (2008).

O formato OBJ caracteriza-se pela geração de dois ficheiros, denominados de file OBJ e file MTL. Estes são responsáveis respetivamente pela geometria e mapeamento de textura. Possui como características ser um formato muito utilizado por aplicações gráficas para comunicação de dados 3D. O formato *Wavefront* OBJ suporta geometria

na forma de vértices, arestas e faces; propriedade de textura, material e grupos de materiais (McHenry *et al.*, 2008).

Apresenta-se na Tabela 11 os formatos de importação e exportação das ferramentas de software analisadas.

Software	Formato de importação	Formato de Exportação
AutoCAD	-	DXF, FBX
Blender	FBX	OBJ e FBX
3DS Max	FBX, DWG e DXF	DXF, FBX e MI
CityEngine	FBX, DXF, OBJ, 3DS	FBX, RIB (Render Man) e MI (NVIDIA'S Mental Ray), OBJ, DAE Collada, .3WS.
Unity 3D	FBX	FBX

Tabela 11 – Interoperabilidade de ferramenta de software.

Na Figura 3 apresentam-se os formatos de ficheiros de dados que podem ser usados para fazer a transferência de dados 3D entre as ferramentas de software analisadas. Estas ferramentas podem associar-se às diferentes etapas de reconstrução de herança cultural digital: modelação tridimensional (AutoCAD); automatização de reconstrução (Blender, 3DS Max, CityEngine) e Renderização (Unity 3D e Mental Ray).

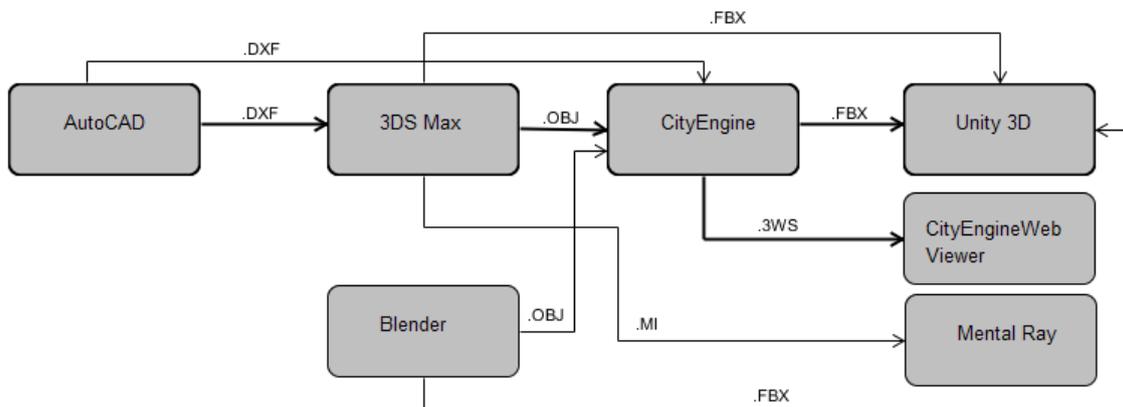


Figura 3 – Interoperabilidade de dados 3D.

Verifica-se que as ferramentas estudadas possuem compatibilidade com os formatos FBX e/ou OBJ.

3.1.2 Modelação tridimensional

No âmbito da herança cultural digital é frequente a reconstrução de sítios ou de objetos em ruínas. Neste caso não é possível recorrer as técnicas baseadas em fotogrametria ou *scanning* e é preciso desenvolver um modelo tridimensional de raiz.

Existem várias ferramentas para modelação tridimensional. Essas ferramentas possuem a capacidade de exportar ficheiros e interagir com outras ferramentas que possuem maiores recursos. Por exemplo, o AutoCAD, oferecido pela Autodesk, detém compatibilidade com 3DS Max, onde é possível melhorar a qualidade das texturas aplicadas. Certifica-se nesta etapa, a capacidade da ferramenta de *software Blender* e 3DS Max terem recursos para modelação tridimensional, além disso, possuem outros recursos como animação do modelo tridimensional e aplicação de texturas.

3.1.3 Automatização de reconstrução

A etapa de automatização de reconstrução de herança cultural digital caracteriza-se por construção automática ou (semi) automática de um ambiente virtual. O principal objetivo consiste em agrupar objetos tridimensionais através de recurso de *script* ou regras gramaticais para automatizar o processo de recriação.

Descrevem-se em seguida as características de automatização das ferramentas estudadas:

- Blender: possui capacidade de desenvolver *script* em *python* e tornar a recriação de ambiente virtual automatizada.
- 3D Studio Max: utiliza o MAXScript e *python script* para automatizar à reconstrução de ambientes virtuais. Ademais, permite criar novas funcionalidades através de *plug-ins* desenvolvido em C# SDK ou C++ SDK.
- CityEngine: caracteriza-se por automatizar reconstrução de ambiente virtual por criação de regras gramaticais, utiliza-se do módulo *CGA Shape*²⁶ para estabelecer parâmetros de reconstrução de um determinado ambiente virtual. Além disso, suporta *python script* para definir automatização de reconstrução.

A seguir detalha-se a automatização de reconstrução no 3DS Max e no CityEngine.

Automatização de reconstrução no 3DS Max

A arquitetura do 3DS Max possui a seguinte característica: arquitetura em *plug-ins* que formam os recursos e funcionalidades do ambiente 3DS Max (Sabir, 2012). A origem de contributo para implementação de *script* na ferramenta 3DS Max surge com a

²⁶ CityEngine, “CGA Shape Grammar Reference”, acessido a 12 de novembro de 2015, em: <http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/cga/basics/toc.html/>

criação do *plug-in DLX-Max Script extension plug-ins*, criado por John Wainwright²⁷, em 1996. O referido autor, disponibiliza curso para MAXScript²⁸.

A arquitetura do 3DS Max permite novas funcionalidade, através da sua capacidade de extensibilidade de *plug-ins*, que podem ser desenvolvidos com diversas abordagens: a) *MAXScript*; b) *PythonScript*; c) *Max SDK*. A capacidade de extensão do 3DS Max possui as seguintes características:

- *Python Script*: o ambiente de desenvolvimento tridimensional 3DS Max oferece módulo para escrever *Python script* em conjunto com *MAXScript*;
- *MAXScript*: consiste em script desenvolvido na camada de alto nível, dentro do ambiente de desenvolvimento;
- *MAX SDK*: atua na camada baixo nível, pode ser utilizado Dot NET em combinação com linguagem de programação C# ou C++.

A API do *Python* do 3DS Max permite a realização de extensão e personalização do *3D Studio Max*, pois, a integração de uma sequência específica de módulos baseado em *Python Script* torna-se mais eficiente o processo de construção de modelação tridimensional, o que permite automatizar o processo de recriação de ambiente virtual.

A API fornece uma ponte entre *MAXScript* e *Python* bibliotecas e as ferramentas de *Python*. Portanto, torna-se possível executar *Python scripts* em combinação com *MAXScript* para adicionar novos recursos. O *MAXScript* foi originalmente desenvolvido como um *plug-in* para Max em 1996 por John Wainwright e Segundo (Sabir, 2012) a *Autodesk* percebeu a importância do Max Script e contratou *John Wainwright* para desenvolver integração junto ao ambiente 3DS Max.

A *AutoDesk* disponibiliza um *plug-in* para permitir *script python* em comunicação com *MAXScript* para realizar a extensão de funcionalidade. Pode concluir-se que o 3DS Max *Phthon* API permite alargar e personalizar 3DS Max, torna-se mais eficiente integrá-lo em uma sequência específica de módulo com base em *python*. Quanto aos produtos relacionados ao 3DS Max, verifica-se o *Mental Ray Standalone* (software de renderização), que possui integração junto ao ambiente de desenvolvimento *3D Studio Max*.

²⁷ Project Gutenberg Self-Publishing Press, “JohnWainwright (Computer Scientist)”, acessido a 23 de novembro de 2015, em: http://self.gutenberg.org/articles/John_Wainwright_%28computer_scientist%29

²⁸ MAXScript 101, Course for MAXScript, acessido a 02 de novembro de 2015, em: <https://vimeo.com/album/1514565/format:detail>.

O objetivo da etapa de automatização de reconstrução é utilizar recursos de extensibilidade que as ferramentas oferecem como recurso e que permite criar um ambiente virtual de reconstrução automática e interoperabilidade de dados 3D, quanto a transição para outras ferramentas que oferecem suporte de renderização, caso do *unity game engine* e *mental ray*.

Automatização de reconstrução no CityEngine

O *software CityEngine* tem um modelo de *pipeline* de ferramentas para criação de modelos 3D através de operações definidas por componentes. Ou seja, a arquitetura do CityEngine possui diversas ferramentas com atribuições específicas para gerar de forma sequencial as etapas necessárias para produzir o modelo tridimensional de cidade virtual.

O CityEngine permite criar modelos 3D de diferentes formas: importar modelos, designado por *static models*, desenvolvidos em outras ferramentas; criação de modelos através da modelação poligonal e a geração de modelos através de código de regras ou *python script*. Este software permite também a entrada de dados por imagens de mapas bidimensionais – mapas geográficos. A principal característica é criar modelo tridimensional de cidades tendo como base a existência de dados de topografia. Através do módulo *CGA Shape* são definidas as regras gramaticais com parâmetros específicos para a construção de cidades virtuais. Os ficheiros CGA são processados em tempo de execução dentro do ambiente CityEngine. Os ficheiros são criados através de programação de regras no ambiente de desenvolvimento, e possuem as seguintes etapas:

- i. Criar um ficheiro com nome da cena (por exemplo, scene.cej);
- ii. Criar ficheiro de regras – ficheiros CGA;
- iii. Desenvolver o conjunto de regras no ficheiro CGA, recorrendo ao editor CGA;
- iv. Definir regra inicial – *@Start Rule*.
- v. Aplicar ficheiro de regras em uma shape 2D.
- vi. Gerar modelos através do botão *Generate*.

Ao criar novas regras com parametrização de um conjunto de regras, utiliza-se a capacidade da ferramenta (módulo *CGA Shape* do CityEngine) de possibilitar a geração de modelos 3D.

Ademais, o CityEngine possui o módulo *Python Script*²⁹, que permite a criação de *script* para automatizar a construção de modelos tridimensionais. Todavia, a interface de *script* CityEngine é baseada em *Jython*, uma implementação Java do *Python*. Para possuir o ambiente de *script* é necessária uma licença junto a ESRI.

3.1.4 Motor de renderização

Após a construção do modelo 3D é possível realizar a sua transposição para um motor de renderização. Entre os motores de renderização analisado, destaca-se o *unity game engine*, *web scene* e *mental ray*, que utilizam o formato de ficheiro FBX, 3WS e MI respetivamente.

O *Mental Ray* caracteriza-se por ser um renderizador de alta qualidade. Utiliza-se a técnica de *bucket rendering*, que permite distribuir a tarefa de renderização para uma única imagem. O *Mental Ray* está integrado ao ambiente de desenvolvimento tridimensional 3DS Max. Acrescenta-se a característica de usar diferentes métodos para calcular luz baseada em imagem.

O CityEngine *Web Scene* caracteriza-se por ser a plataforma de renderização de modelos 3D desenvolvidos em camadas. De entre as funcionalidades, destaca-se a possibilidade de percorrer a cena utilizando visão panorâmica e selecionar camadas específicas de visualização. Utiliza a biblioteca OpenGL para apresentar o modelo virtual para interação.

O Unity 3D possui como característica principal e diferencial um ambiente de desenvolvimento para criar jogos e experiências interativas tridimensional. O motor de jogos possui capacidade e suporte para multiplataforma. A plataforma possibilita desenvolver *script* em C# e *UnityScript*.

Após projeto construído, torna-se possível exportar para diversas plataformas atendendo o requisito de portabilidade, pois o lema consiste: “crie uma vez e implemente onde quiser”³⁰. Entre as vantagens do ambiente de desenvolvimento do Unity 3D, destaca-se o motor de jogos, que permite desenvolver visualização em tempo real pela web. A justificação para a sua utilização, verifica-se na descrição de (Merlo *et*

²⁹ CityEngine, “Python Scripting Notes and Changelog”, acessado a 07 de julho de 2016, em: http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/python/python_iface.html

³⁰ Unity – Multiplatform, “Publish your game to over 10 platform”, acessado a 13 de novembro de 2015, em: <http://unity3d.com/pt/unity/multiplatform/>

al.; 2012): “Motor de jogo Unity 3D, porque este aplicativo oferece suporte em tempo real de renderização de técnicas avançadas”.

Uma das características do Unity 3D consiste em interação com ambiente de desenvolvimento *Microsoft Visual Studio*, onde torna-se vantajoso e recomendável a utilização de *scripts C#* para definir comportamento de objetos no cenário tridimensional.

3.1.5 Ferramentas adotadas

A Tabela 12 resume as características das ferramentas de software estudadas tendo em atenção as etapas necessárias para a reconstrução digital de espaços históricos (modelação 3D, automatização de reconstrução e motor de *rendering*), o tipo de licença de utilização, a interoperabilidade de dados 3D, fontes de formação e a modelação de personagens virtuais e animação.

Estas ferramentas possuem as características necessárias para as várias etapas do processo de criação do cenário de Mértola Virtual, compatibilidade de transferência de dados 3D entre elas, soluções de modelação semi-automática de reconstrução de ambiente urbano virtual e, ou são gratuitos, ou têm licenças académicas que permitem a sua utilização com custos reduzidos.

Além das ferramentas já mencionadas, foi necessário recorrer à plataforma ArcGIS para o processamento de terreno. A plataforma ArcGIS é um sistema de informação geográfica (SIG). O conceito de SIG segundo (Cosme, 2012, p. 43): “Um SIG é um sistema de captura, armazenamento, análise, gestão e apresentação de informação georreferenciada, ou seja, ligada a uma localização espacial. (...) As aplicações SIG são ferramentas que permitem realizar visualização, exploração e análises espaciais. Cumulativamente, permitem a edição de mapas e produção de *outputs* de operações efetuadas sobre a informação”.

O ArcGIS oferece suporte para o pipeline de geração de terreno, combinando a recolha de dados do modelo digital do terreno da fonte *Open Topography* e a imagem de satélite do Google Earth para textura. Os dados adquiridos são processados, em ambiente ArcGIS Desktop em computador local, e geram como resultado diversos ficheiros de *output*.

Descrição	AutoCAD	Blender	3DS Max	CityEngine	Unity 3D
Modelação 3D	Sim	Sim (importação)	Sim (importação)	Sim (importação)	Sim (importação)
Automatização de reconstrução	Não	Python script	MAXScript Python script	Grammar Rule Python script	C# UnityScript
Motor de rendering	Não	Sim	Mental Ray	CityEngine Web Scene	Unity 3D Game Engine
Domínio	Proprietário – possui licença académica	Público	Proprietário – possui licença académica	Proprietário – possui licença académica de um ano.	Proprietário – possui versão académica (<i>Personal Edition</i>).
Interoperabilidade de dados 3D	Sim	Sim.	Sim.	Sim.	Sim.
Fontes de formação	Fontes diversas de aprendizado - Sítio principal ³¹	Fontes diversas de aprendizado - Sítio principal ³² .	Fontes diversas de aprendizado – sítio principal ³³ .	Fontes diversas de aprendizado – sítio principal ³⁴ .	Fontes diversas de aprendizado – sítio principal ³⁵ .
Modelação de Personagens Virtuais e animação.	Não	Sim.	Sim -	Não.	Sim.

Tabela 12 – Ferramentas de software a adotadas.

Foram utilizadas as seguintes versões destas ferramentas:

- AutoCAD: versão 2016 académica.
- Blender: versão 2.73.
- 3DS Max: versão 2015 académica.
- ArcGIS (ArcMap): versão 10.3.1 – versão de avaliação de sessenta dias.
- Gimp: Versão 2.8.16.
- CityEngine: versão 2014.0 académica de um ano.

³¹ Autodesk, “AutoDesk AutoCAD”, acessado a 12 de julho de 2016, em: <https://knowledge.autodesk.com/support/autocad/learn-explore/#?sort=score>

³² Blender, “Blender Reference Manual”, acessado a 19 de novembro de 2015, em: <http://www.blender.org/manual/contents.html>

³³ Autodesk, “3DS Max 2016 Help”, acessado a 19 de novembro de 2015, em: <http://help.autodesk.com/view/3DSMAX/2016/ENU/>

³⁴ ESRI City Engine, “CityEngine Tutoriais”, acessado a 19 de novembro de 2015, em: <https://desktop.arcgis.com/en/cityengine/latest/tutorials/introduction-to-the-cityengine-tutorials.htm>

³⁵ Unity Documentation, “Unity Manual”, acessado a 19 de novembro de 2015, em: <http://docs.unity3d.com/Manual/index.html>

- Photoshop: versão 2015 de avaliação de um mês.
- Unity 3D: versão 5.3.3.

A plataforma CityEngine exige alta capacidade de memória e placa de vídeo para processar o ambiente urbano virtual. Torna-se necessário no mínimo 1GB de placa de vídeo VRAM dedicada, pois todas a textura do modelo 3D são carregadas nesta placa dedicada. É utilizada a biblioteca OpenGL, que usa os recursos de hardware da placa de vídeo.

A Tabela 13 apresenta as configurações mínimas de *hardware* para suportar o processamento e eficácia na visualização do modelo tridimensional.

CPU	2 GHz dual-core CPU mínimo (Pentium 4- compatível com Intel/AMD)
Memória RAM	8GB ou mais (16 GB recomendável)
Disco	1GB ou mais.
Placa de vídeo	1GB dedicado.

Tabela 13 – Requisito de hardware para CityEngine.

3.2 Caso de estudo – Mértola Virtual

O cenário virtual urbano a desenvolver consistiu na reconstrução do espaço histórico da vila de Mértola (município do distrito de Beja) no período de ocupação Islâmica, entre os séculos VII e XIII.

Na vila de Mértola durante o período islâmico constata-se em registos arqueológicos a existência de um conjunto de casas localizadas num bairro, com denominação de bairro Islâmico ou Alcáçova, que antes da conquista da cidade pela ordem dos cavaleiros, era ocupado por cerca de vinte moradias (Macias, 1996). As moradias eram padronizadas no estilo clássico mediterrânico: em volta de um pátio interior, organizavam-se os vários compartimentos da casa; o telhado da casa estava inclinado para o pátio interior. Na Figura 4 apresenta-se, à esquerda, uma imagem das ruínas e, à direita, a planta de uma das casas do bairro islâmico (Torres, 2008).



Figura 4 – Ruínas atuais do bairro islâmico e planta (Torres, 2008).

Além da Alcáçova, há ainda registo de outras estruturas desta época. Martínez et al. (2009) referem a muralha da Madina, estrutura intramuros da cidade que possuía habitações no período Islâmico. Macias (1996) menciona outra área adicional, designado de Arrabalde.

No trabalho de (Macias, 1996b), relata-se um levantamento de uma área de aproximadamente quatro mil metros quadrados. Em parte desta área foram identificadas 15 casas, que estão representadas na planta da Figura 5, que integram a Alcáçova.



Figura 5 – Conjunto de residências (Macias, 1996b).

Existem gravuras de Mértola de uma época posterior que são da autoria de Duarte de Armas, que foi enviado por Dom Manuel I para catalogar as fortalezas de Portugal no século XVI. A Figura 6, da sua obra Livro das Fortalezas, representa Mértola nesta época.



Figura 6 – Mértola fortificada (Armas, 1509).

Em comparação ao registo de Duarte de Armas, a Figura 7 mostra uma vista parcial recente de Mértola.



Figura 7 – Vista parcial de Mértola (Torres, 2008).

A recriação dos modelos arquitetónicos de Mértola tiveram por base a consulta a desenhos e esboços do espaço histórico, a artigos de arqueólogos e historiadores, fotografia e registos históricos.

Para o modelo de terreno obtiveram-se dados *raster* com o modelo digital do terreno (*digital elevation model*), dados vetoriais para o processamento de dados geográficos e imagens para textura do terreno.

3.3 Conclusão

Este capítulo apresentou as ferramentas analisadas para o desenvolvimento da reconstrução do espaço histórico da vila histórica de Mértola no período de ocupação Islâmica. Estas ferramentas têm em atenção a recriação de modelos 3D de estruturas que já não existem mais ou que se encontram em ruínas, situação do presente caso de estudo, onde os edifícios do período islâmico são recriados com base em fontes de dados arqueológicos.

Indicaram-se as ferramentas utilizadas no desenvolvimento do trabalho e os requisitos básicos quanto a *hardware* suficiente para alcançar eficácia e eficiência no processamento de produção do cenário virtual.

No capítulo seguinte descreve-se a construção do modelo do terreno, a implementação do ambiente urbano virtual semi-automatizado por *procedural rule CGA* e a integração de objetos 3D desenvolvidos a partir da modelação tridimensional.

Capítulo 4

Pipeline de geração do ambiente virtual

O presente trabalho tinha como objetivo viabilizar a reconstrução de um espaço histórico, em Mértola, de modo a permitir a sua exportação para a plataforma Unity 3D, para criar um ambiente virtual interativo com personagens virtuais.

Como já foi referido, neste trabalho utilizou-se o CityEngine como plataforma de desenvolvimento do ambiente virtual urbano. No processo de desenvolvimento consideraram-se duas etapas: a modelação do terreno e a modelação dos edifícios.

As principais etapas de desenvolvimento do ambiente virtual e a integração das várias componentes e ferramentas utilizadas em cada uma delas, estão representadas na Figura 8, e serão descritas nas secções seguintes.

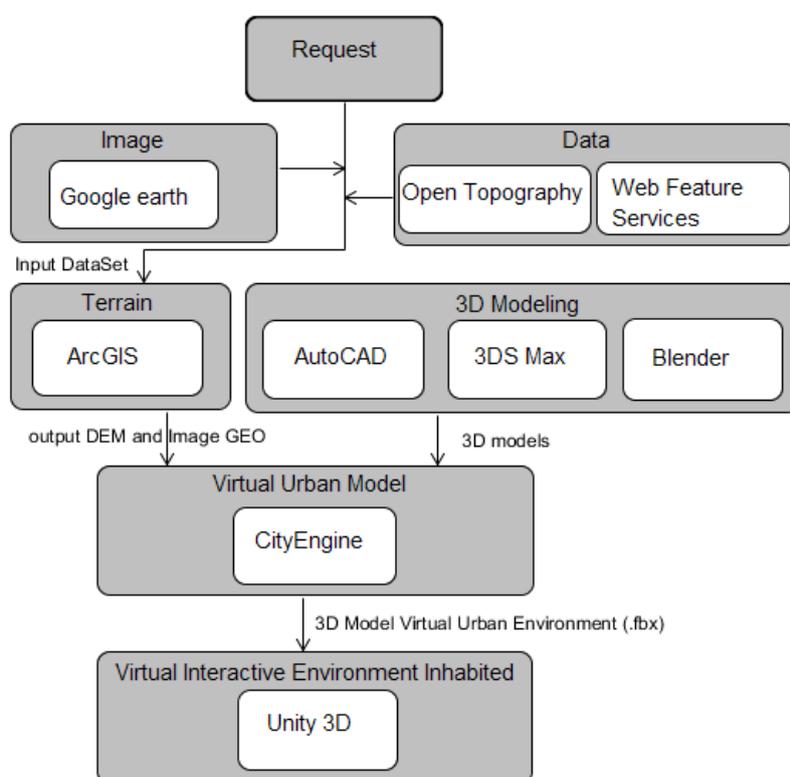


Figura 8 – Etapas para produção do ambiente urbano virtual.

4.1 Geração do terreno

A presente secção descreve a obtenção de dados para a geração do terreno do ambiente virtual urbano e, posteriormente, a sua importação no CityEngine. Utilizou-se o sistema de informação geográfica ArcGIS, em combinação com outras ferramentas, para a recolha e processamento dos dados do terreno e de imagem georreferenciada, que serviram como fonte de dados de relevo e textura para importação no ambiente CityEngine. As várias etapas deste processo estão representadas na Figura 9, onde se identificam as seguintes tarefas principais:

- **Entrada de dados** (*Input Dataset*), que consistiu em: a) recolher os dados digitais do terreno através do sítio *Open Topography*; b) recorrendo à componente ArcCatalog do ArcGIS, ligar ao serviço Web da Direção Geral do Território³⁶, que fornece dados vetoriais do território nacional, para gerar uma *shapefile*³⁷, delimitando uma área correspondente aos dados vetoriais importados.
- **Geração de imagem de textura** (*Image generation*): exportar a *shapefile*, criada no passo anterior, para o Google Earth Pro³⁸, no formato KML³⁹, para fazer o download de uma imagem de alta resolução, onde estão assinalados os vértices da *shapefile* (pontos de controlo).
- **Processamento de dados e imagem** (*Data and Image Processing*): a) com a componente ArcMap, do ArcGIS, usando os pontos de controlo assinalados na imagem, fazer a georreferenciação e também o recorte da imagem de modo a que a textura corresponda apenas à área de interesse; b) com a componente ArcToolbox do ArcGIS, gerar os ficheiros relacionados ao modelo digital do terreno, que contém a informação do relevo para o CityEngine.

Em seguida descrevem-se com mais detalhes cada uma destas tarefas.

³⁶ DG Território, “Guias de Apoio – Guia de apoio à utilização de serviços WFS – ArcGIS”, acedido a 28 de março de 2016, em: <http://mapas.dgterritorio.pt/>.

³⁷ A informação do objeto *shapefile* é geograficamente referenciada, sendo este objeto composto por vários ficheiros de extensões diferentes (Cosme, 2012).

³⁸ Google Earth, acedido a 02 de junho de 2016, em: <https://www.google.com/earth/>

³⁹ Keyhole Markup Language, “What is KML?”, acedido a 06 de junho de 2016, em: <https://developers.google.com/kml/>

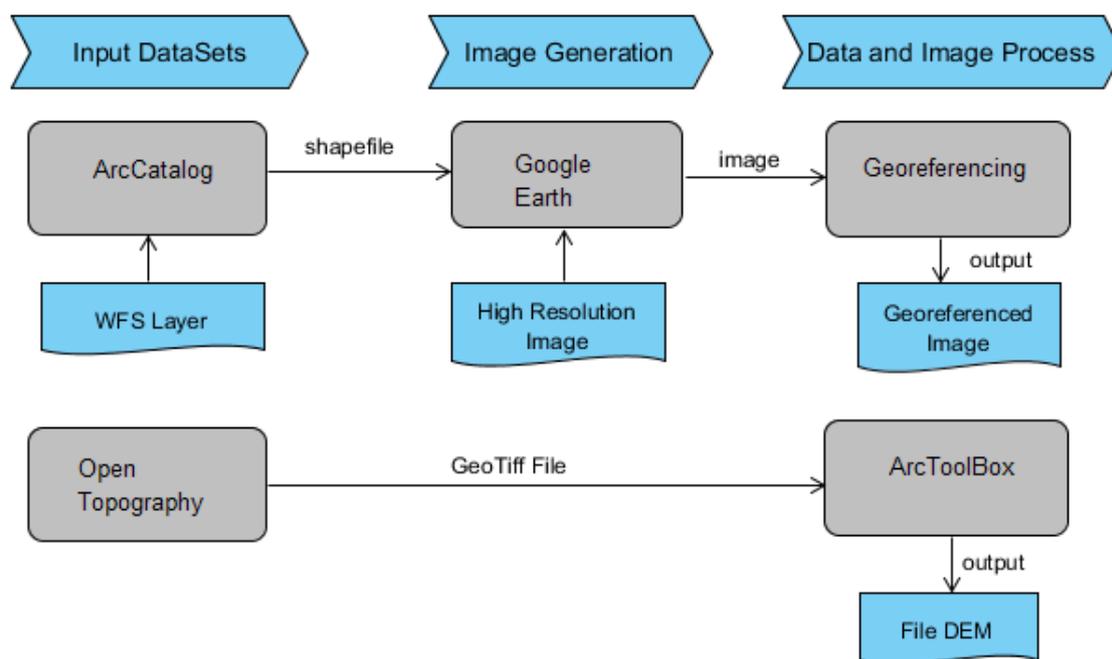


Figura 9 – Etapas do processamento de informação geográfica.

Entrada de dados

A Direção Geral do Território dispõe do modelo digital de terreno (*raster dataset*) para o território nacional. Contudo, a aquisição destes dados tem custos e o orçamento para a zona de Mértola corresponderia a 68,80 euros (sessenta e oito euros e oitenta cêntimos). Optou-se, por isso, por recorrer ao sítio da *Open Topography*⁴⁰, que disponibiliza estes dados gratuitamente, bastando apenas um registo no sítio para *download* dos dados da área de interesse.

A Figura 10 mostra a interface do *Open Topography* para aquisição do modelo digital de terreno, onde foram indicadas manualmente, no sistema de coordenadas WGS 1984, os limites que correspondem a uma área de interesse com, aproximadamente 1 Km², centrada na região intramuro de Mértola. Utilizou-se o Google Earth para fazer medições e obter as coordenadas geográficas adequadas para se selecionar a área de interesse.

O *Open Topography* exporta um ficheiro no formato GeoTiff com a área selecionada. Este ficheiro será usado posteriormente na tarefa de processamento de dados e imagem.

⁴⁰ Open Topography, “Find Lidar Topography Data”, acessido a 02 de junho de 2016, em: <http://opentopo.sdsc.edu/datasets>



Figura 10 – Aquisição de modelo digital do terreno.

A etapa seguinte da tarefa de entrada de dados é a geração de uma *shapefile*, que delimite a área relativa aos dados *raster* importados. Para este efeito, utilizou-se, num primeiro passo, a componente ArcCatalog, do ArcGIS, para estabelecer a ligação ao *Web Feature Services* (WFS) da Direção Geral do Território, que fornece dados vetoriais do território nacional. O WFS permite a manipulação e consulta de dados com base na diretiva INSPIRE, que recomenda boas práticas no uso dos serviços de dados geográficos definidos pelo OGC (*Open Geospatial Consortium*). Patrício et al. (2015) referem que “os *web services* são regras que normalizam a forma de trocar informação geográfica de forma aberta e transparente”. Além dos WFS, há outros tipos de serviços como: *Web Map Service* (WMS), *Geograph Markup Language* (GML) e *Web Coverage Service* (WCS)

Foi necessário instalar a extensão do ArcGIS *Data Interoperability*, módulo de interoperabilidade de dados disponível no sítio da ESRI⁴¹. Após a instalação, utiliza-se o ArcCatalog para definir os parâmetros de configuração do ArcGIS *Data Interoperability*⁴² para conexão à base de dados de distribuição de dados geográficos em formato vetorial da Direção Geral do Território. Estabelecida a ligação, podemos selecionar interactivamente a área de interesse do concelho de Mértola. Com base nestes

⁴¹ ESRI, “ArcGIS Data Interoperability”, acedido a 10 de junho de 2016, em: <http://www.esri.com/software/arcgis/extensions/datainteroperability>

⁴² DG Território, “Guias de Apoio – Guia de apoio à utilização de serviços WFS – ArcGIS”, acedido a 28 de março de 2016, em: <http://mapas.dgterritorio.pt/>.

dados vetoriais é criada a *shapefile* denominada *points_of_control*, que delimita a área de interesse e cujos vértices são designados por pontos de controlo.

Geração de imagem da textura

Para obter uma imagem que sirva de textura para o modelo do terreno, criou-se a partir da *shapefile* no passo anterior um ficheiro de formato KML, recorrendo à componente ArcMap do ArcGIS para exportar. O Google Earth Pro importa este ficheiro e assinala sobre a imagem os pontos de controlo da *shapefile*. Esta ferramenta permite fazer o download da imagem com alta resolução. Esta imagem abrange uma área maior do que a *shapefile*, o que vai requerer o seu recorte no passo seguinte do processamento.

A Figura 11 apresenta o resultado da aquisição da imagem de alta resolução pela software Google Earth (em destaque estão os quatro pontos de controlo criado através do *shepefile*).

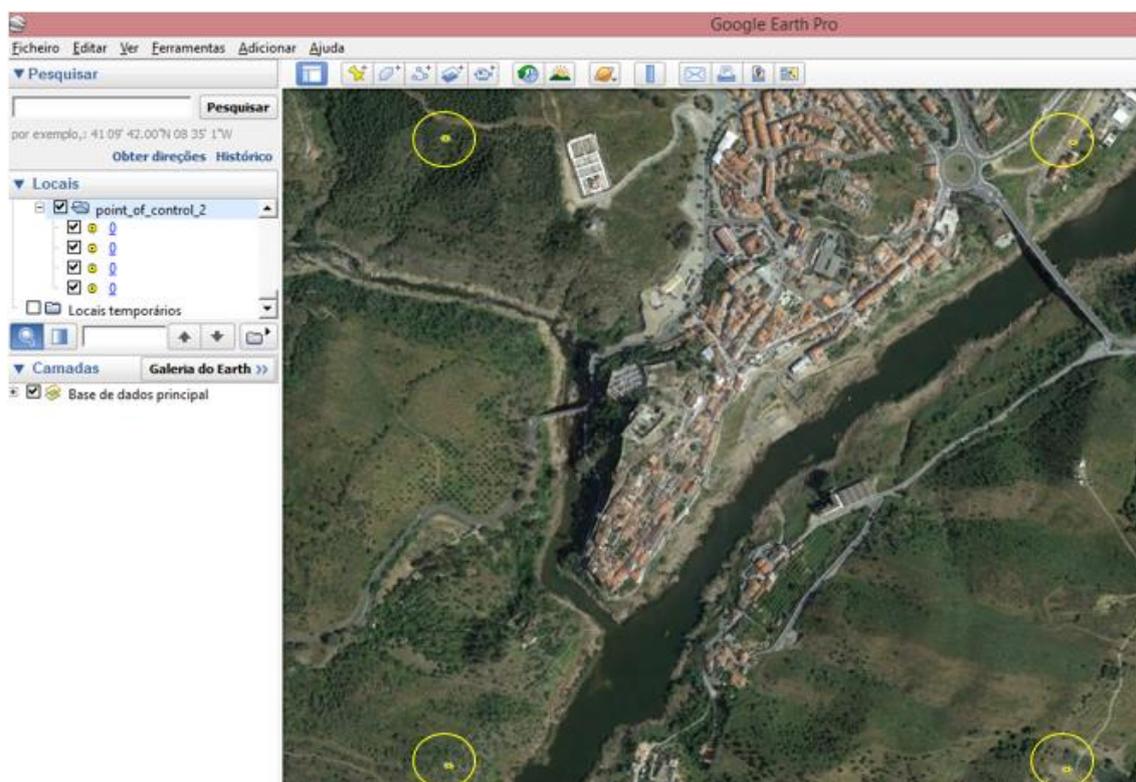


Figura 11 – Aquisição de imagem de alta resolução.

Processamento de dados e imagem

A tarefa de processamento de dados e imagem começa pelo georreferenciamento da imagem no ArcMap. Importa-se a imagem gerada pelo Google Earth Pro e utiliza-se a ferramenta *georeferencing* para fazer a georreferenciação da imagem através dos

pontos de controle da *shapefile points_of_control* criada. Esta imagem é a seguir validada com a operação *rectify*. O recorte da área de interesse é realizado pela ferramenta *mosaic to new raster*. O *mosaic dataset* constitui num polígono convertido em *feature*, que são informações que são armazenada numa *Geo data base*⁴³. Este processo recorta a imagem de forma a termos a textura apenas com a área contida na *shapefile* ponto de controlo criado (área de interesse).

Em seguida, para o processamento de terreno, realiza-se a importação para o ArcMap do ficheiro de dados obtido através *Open Topography (output_srtm.tif)* para gerar os ficheiros de *output*. Na componente ArcToolBox, utiliza-se o recurso *Spatial Analyst Tools* para processar valores para o modelo digital de terreno. Nesta operação é usado uma camada como o nome do ficheiro (*output_srtm.tif*). A esta camada é associada automaticamente os valores máximos e mínimos das altitudes identificados no ficheiro e assinalado na Figura 12. Estes valores são usados para o mapeamento da esquematização em forma de cinzento do *heightfile*.

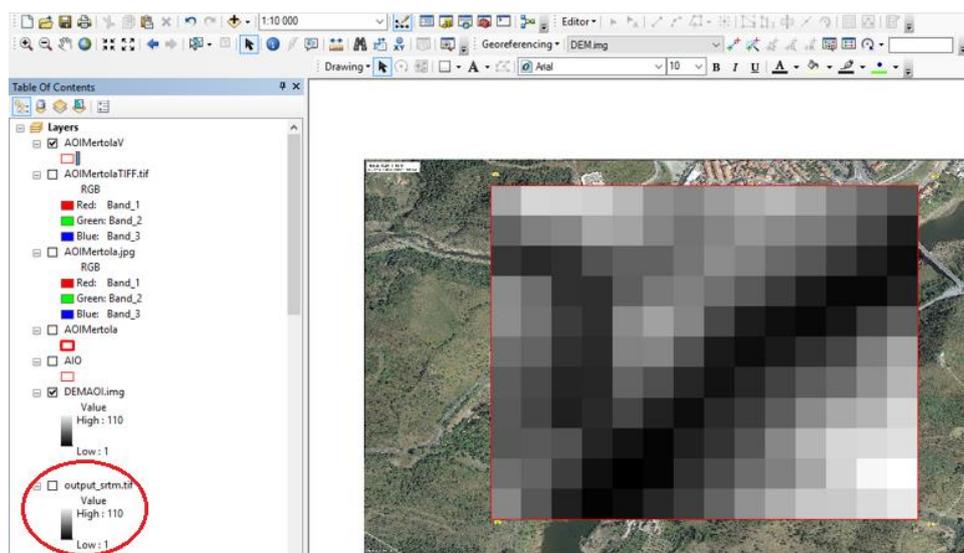


Figura 12 – Modelo digital do terreno processado no ArcMap.

O resultado do processo do *pipeline* ArcGIS gera diversos ficheiros (Figura 13) para imagem georreferenciada e para o modelo digital do terreno, ambos os ficheiros são exportados para a estrutura do projeto CityEngine, o que possibilita criar uma cena com *map layer terrain (heightmap file)* e ficheiro de textura com a imagem georreferenciada (*Texture file*).

⁴³ ArcMap, “Creating a mosaic dataset”, acessado a 01 de julho de 2016, em: <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/creating-a-mosaic-dataset.htm>

Através dos ficheiros gerados, cria-se as camadas (*layers*) de mapa, inserindo os ficheiros que correspondem aos dados manipulados na plataforma ArcGIS (ArcMap). A Figura 13, verifica-se os *inputs* dos ficheiros exportados do pipeline ArcGIS, gerado com o processamento de dados geográficos, lado esquerdo, e imagem georreferenciada e ficheiro em cima e lado direito, imagem de altura do terreno e ficheiros em cima.

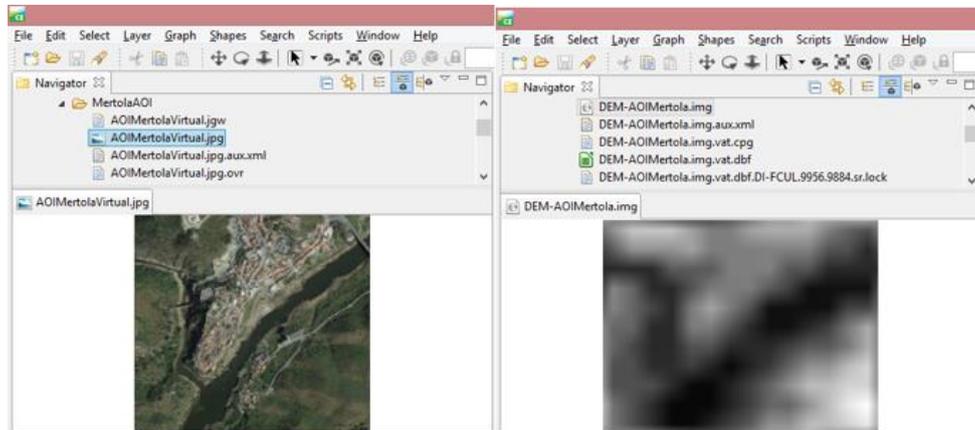


Figura 13 – Imagem georreferenciada e modelo digital do terreno.

O diagrama de atividades na Figura 14 clarifica o fluxo de tarefas realizada inerente a recolha de dados referenciado na Figura 9:

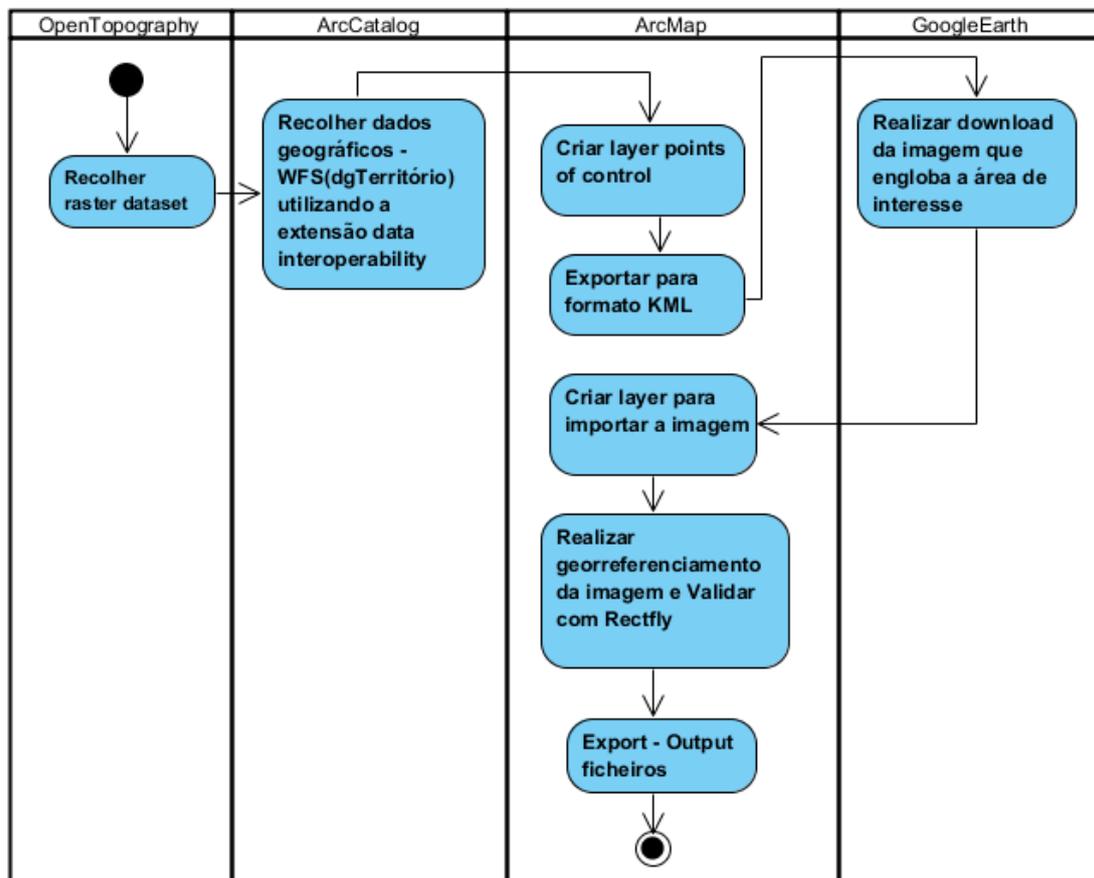


Figura 14 – Fluxo de atividade para recolha de dados.

O anexo A apresenta de forma detalhada a tarefa de processamento de dados e imagem realizada no ArcGIS.

Importação de dados do terreno para o CityEngine

A criação de terreno no CityEngine envolve a importação dos ficheiros da imagem georreferenciada que lhe vai servir de textura (Figura 13, à esquerda) e do modelo digital do terreno (*heightmap*) (Figura 13, à direita) e, através do recurso *new map layer*. A Figura 15 corresponde a criação do terreno na cena virtual.

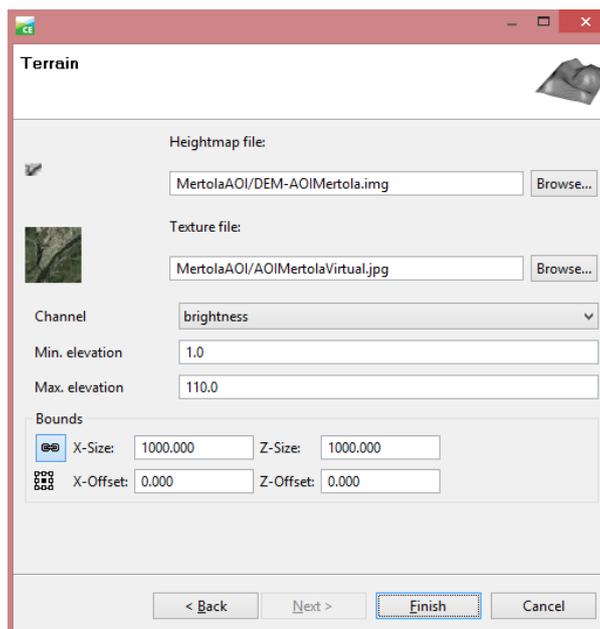


Figura 15 – Criação de camada terreno.

Após adicionar os ficheiros de modelo digital do terreno e imagem de textura a camada *map layer* é criada.

4.2 Geração de ruas e de *shapes* para os modelos arquitetónicos

O CityEngine oferece duas opções para geração de uma rede de ruas e de blocos limitados por esta ruas: *design street manually* e *street networks grow*. Por uma questão de eficiência, optou-se por usar a geração automática. Esta opção de construção automática usa o recurso *Grow Street* que requer a associação a um ficheiro *heightmap*, com o modelo digital de terreno, com um ficheiro obstáculo. Este último ficheiro impede que, no caso em estudo, se gerem ruas sobre o leito dos rios.

O ficheiro obstáculo contém uma imagem que através do brilho vai distinguir duas zonas: uma onde se podem colocar ruas e outra que não pode ser atravessada por ruas. No caso particular do caso de estudo de Mértola Virtual, este ficheiro tem os leitos

dos rios Guardiana e Oeiras a preto e a restante área a branco (Figura 16). Este ficheiro resultou de editar uma cópia da imagem georreferenciada da textura do terreno, usando um editor de imagem como, por exemplo, as aplicações Photoshop⁴⁴ ou Gimp⁴⁵. Com o recurso *new graph layer* cria-se uma camada para guardar este ficheiro obstáculo e, nesta camada, define-se o valor do atributo que impede o desenho de ruas. Este atributo tem um valor booleano definido em função do valor do brilho, por exemplo, “*attr obstacle = brightness < 0.5*”. Deste modo não são desenhadas ruas sobre a cor preta, porque tem brilho inferior a 0.5.



Figura 16 –Imagem do ficheiro obstáculo

Após associar os ficheiros de *heightmap* e de obstáculo ao recurso *Grow Street*, podemos gerar automaticamente as ruas escolhendo um tipo de padrão: *organic*, *raster* ou *radial*. No caso de estudo Mértola Virtual foi usado o padrão *organic*, porque gera todas as ruas com a mesma largura e de forma mais aproximada ao que é referido na literatura (Martínez et al., 2009), nomeadamente, “ruas estreitas e de terra batida”. Na Tabela 14 apresentam-se os parâmetros escolhidos.

Atributos	Valor
shapeCreation	True
streetWidth	3
streetOffset	0
sidewalkWidthLeft	1
sidewalkWidthRight	1
Precision	0.1

Tabela 14 – Definição de parâmetros para street.

⁴⁴ Adobe Creative Cloud, “Photoshop CC”, acedido a 20 de fevereiro de 2016, em: <http://www.adobe.com>

⁴⁵ Gimp, acedido a 04 de julho de 2016, em: <https://www.gimp.org/>

Após a geração das ruas é necessário aplicar funções de alinhamento para melhor ajustar os objetos criados ao relevo do terreno. Os tipos de alinhamento utilizados foram os seguintes:

- 1) *Align graph to terrain*: utiliza-se esta opção para alinhar as ruas e *shapes* criadas com o terreno. Os parâmetros de configuração deste alinhamento são o *heightmap* (que tem informação sobre a elevação do terreno) e *offset* (posicionamento dos gráficos a alinhar em relação a camada do terreno).
- 2) *Align the terrain to shapes*: esta opção realiza a operação inversa da anterior, ou seja, faz o alinhamento do terreno em relação às *shapes*. Ao alinhar o terreno, o *heightmap* original do terreno não é modificado. Os dados de variação da elevação são armazenados noutro ficheiro de imagem (*elevationDelta*) numa subpasta (com o mesmo nome da cena) na pasta de dados do projeto (ver Figura 17).

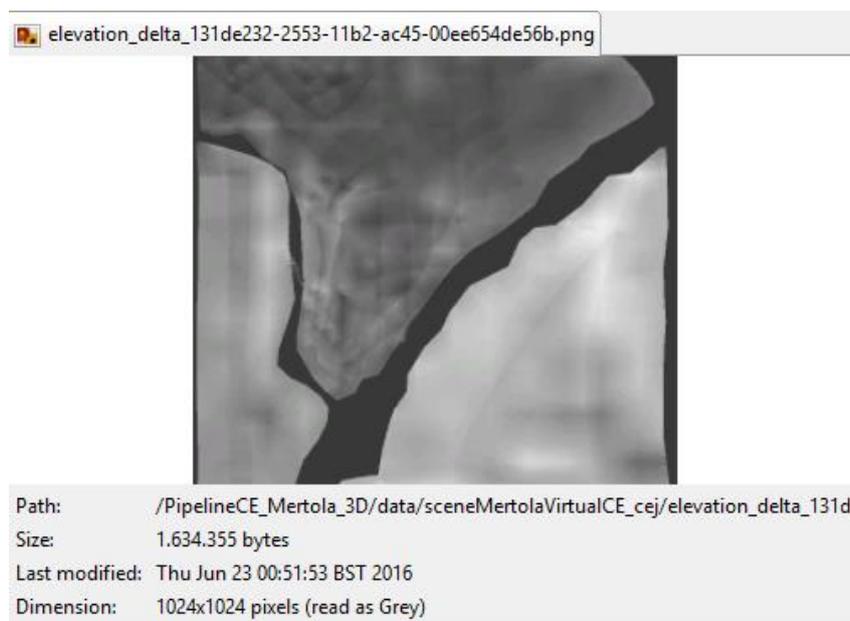


Figura 17 – Imagem de *ElevationDelta* do terreno.

A principal característica do ficheiro *elevationDelta*, que é criado pela plataforma, consiste em armazenar informações quanto ao processo de alinhamento de objetos gráficos ao terreno⁴⁶. Na Figura 17, apresenta-se a imagem criada, assinalando as mudanças efetuadas: a cor cinza, significa que não houve mudança no relevo, a cor preta, que ficou mais baixo e a cor branca que ficou mais alto.

⁴⁶ ESRI CityEngine, “Tutorial 2: Terrain and dynamic city layouts”, acessado a 04 de julho de 2016, em: <https://desktop.arcgis.com/en/cityengine/latest/tutorials/tutorial-2-terrain-and-dynamic-city-layouts.htm>

O resultado obtido depende do atributo *elevation* definido na camada de terreno, que fica ativo quando é selecionado o campo *enable elevationDelta*. Neste caso ao atributo *elevation* foi atribuído o seguinte:

```
attr elevation = elevationDelta + map_01(brightness, 1.0, 110.0)
```

A Figura 18 mostra, à esquerda, a *shape street* sem textura, usando os valores da Tabela 14, e; à direita, a *shape street* com aplicação de textura definida através de um ficheiro de regras, o que será explicado mais à frente.

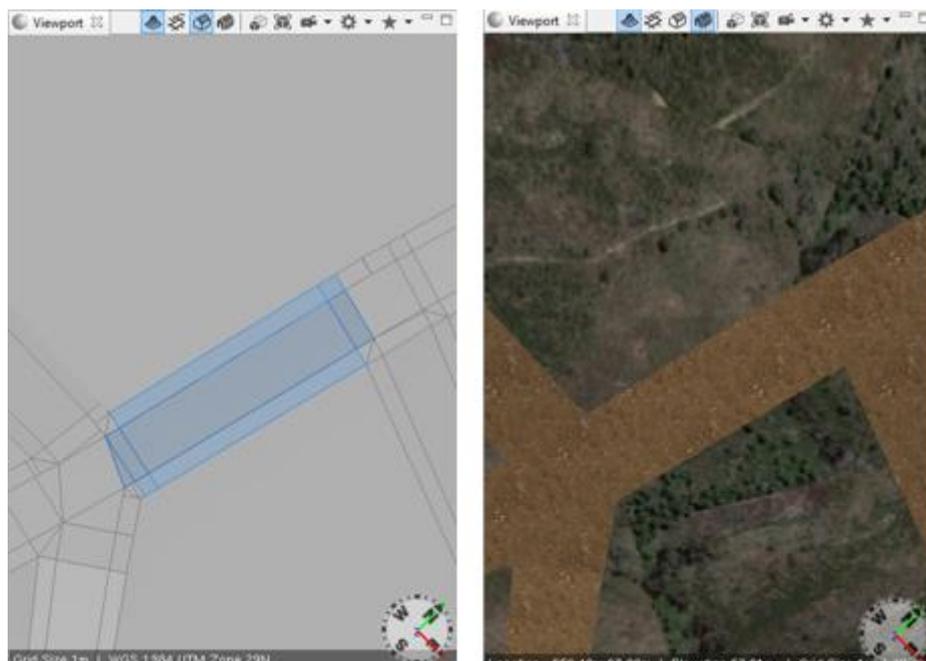


Figura 18 – Visualização parcial da rua com e sem textura.

4.3 Modelação tridimensional

A modelação das edificações do caso de estudo de Mértola Virtual recorreu às técnicas de modelação de raiz, uma vez que se trata de estruturas que não existem na atualidade. Mesmo a muralha atual, não coincide exatamente com a muralha do período em estudo. A informação necessária para a modelação foi recolhida em artigos publicados sobre o assunto e em material disponibilizado pela equipa do campo arqueológico de Mértola.

O ambiente virtual foi criado no CityEngine integrando modelos construídos com ferramentas de modelação tridimensional, como o AutoCAD, o Blender e o 3DS Max, designada por modelação manual por Haegler (2009), e a modelação realizada no

CityEngine, modelação procedimental e modelação poligonal complementada com recurso a regras.

Apresentam-se em seguida as diferentes técnicas de modelação usadas.

4.3.1 Modelação tridimensional manual

No desenvolvimento de Mértola Virtual reutilizaram-se modelos construídos no AutoCAD (algumas casas e a Torre do Rio) e no Blender (uma casa). Estes modelos resultaram de projetos de avaliação de disciplinas de licenciatura e mestrado do Departamento de Informática e, além de acrescentarem mais edificações ao ambiente virtual, permitiram também testar o *pipeline* de construção de modelos virtuais apresentado neste documento. Adicionalmente, foram criados no 3DS Max os modelos de quinze casas da Alcáçova (ver Figura 5), algumas das quais com três níveis de detalhe.

Devido a problemas de incompatibilidade de exportação de texturas nas versões instaladas do AutoCAD e do CityEngine, exportaram-se os modelos do AutoCAD para o 3DS Max para aperfeiçoar as texturas e depois exportaram-se para o CityEngine. Apesar dos modelos criados no Blender poderem ser exportados diretamente para o CityEngine, optou-se por os exportar primeiro para o 3DS Max para que os ficheiros relativos às texturas tenham a mesma organização, independentemente da ferramenta usada para os desenhar.

A Figura 19 apresenta o *pipeline* de integração de objetos tridimensionais até à sua exportação para o Unity 3D. Assinalam-se na figura os formatos de ficheiro usados em cada transição.

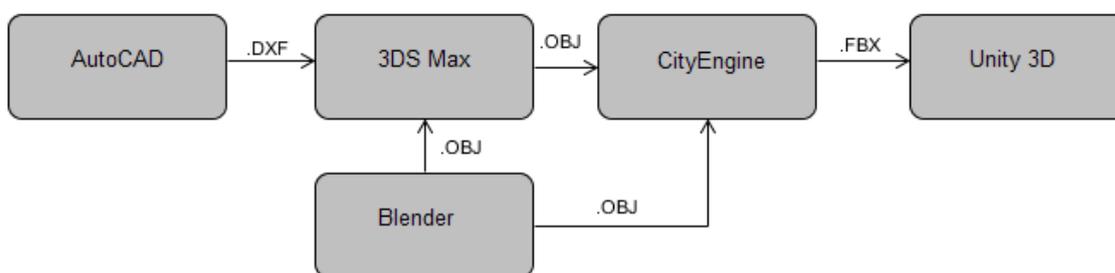


Figura 19 – Visão global do pipeline modelação.

4.3.1.1 Níveis de detalhe

Para permitir a geração de imagens em tempo real no Unity 3D é necessário reduzir, sempre que possível, o número de polígonos. A criação de modelos com

diferentes níveis de detalhe, em inglês *Levels of detail* (LOD), permite usar representações com diferentes níveis de complexidade consoante a posição da câmara. A combinação eficiente de níveis de detalhe torna-se fundamental para obter um melhor resultado. Para esse efeito, a modelação das casas da Alcáçova (ver Figura 5) realizada no 3DS Max teve em atenção a definição dos seguintes níveis de detalhe:

- LOD_0: polígonos correspondentes às superfícies das paredes exteriores, das paredes do pátio interior e telhado a que foram aplicadas texturas, recorrendo ao UVW Map do 3DS Max.
- LOD_1: Paredes exteriores e as paredes do pátio interior com abertura para a porta, paredes interiores, chão do pátio com textura e telhado com textura.
- LOD_2: relativamente ao LOD_1, acrescentou-se o telhado modelado com polígonos, utilizando o recurso do *plug-in Atiles*⁴⁷, que permite criar telhados com elevado nível de detalhe de maneira automatizada sobre a superfície geométrica do telhado. Em seguida, a textura foi aplicada com o recurso UVW Map, que posicionou com eficiência as coordenadas de x, y e z para o formato do telhado.

O LOD_0 consiste no nível de detalhe mais baixo, portanto, com menos polígonos. Ao aumentar o nível de detalhe, LOD_1 e LOD_2, aumenta o número de polígonos e ao mesmo tempo o detalhe do objeto criado (Figura 20).

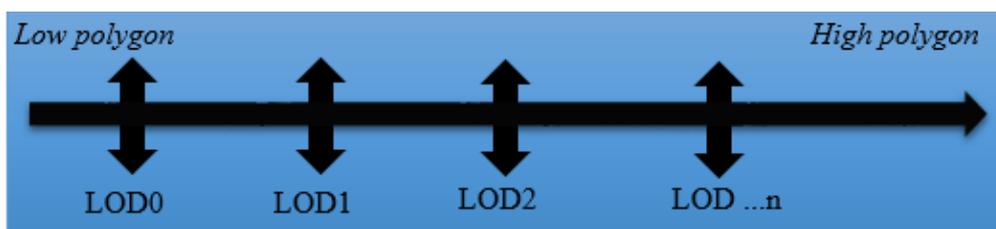


Figura 20 – Contagem de polígonos para LOD.

Todas as casas da Alcáçova foram modeladas com o nível LOD_0 e apenas as casas identificadas na planta com XI, XII e XV foram desenhadas também nos níveis LOD_1 e LOD_2. A Figura 21 mostra à esquerda todas as casas modeladas: as casas XI, XII e XV com LOD_2 e as demais com LOD_0. A visão global dos modelos 3D das casas da Alcáçova encontra-se no anexo E.

⁴⁷ Aviz Studio, “ATiles”, acessado a 11 de junho de 2016, em: <http://www.avizstudio.com/tools/atiles/>

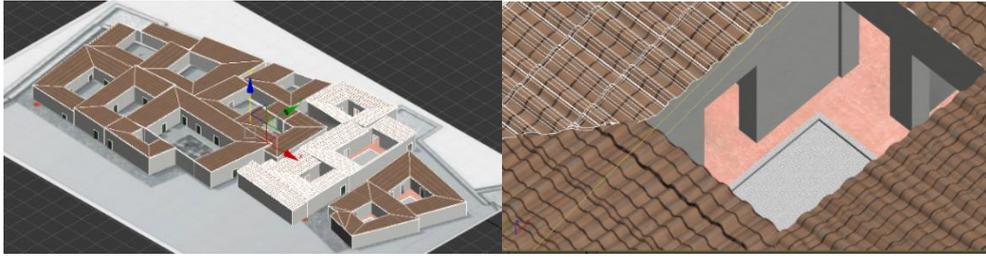


Figura 21 – Modelação das casas em 3DS Max.

O processo de modelação consistiu-se nas seguintes etapas:

- 1) Criar uma camada (*layer*) onde se coloca a imagem da planta completa da Alcáçova.
- 2) Para cada casa, criar as hierarquias das camadas para cada nível de detalhe (por exemplo, em baixo apresenta-se a hierarquia com as camadas para a casa 11 no LOD_2):
 - a. Casa11_parede_externa_lod_2
 - b. Casa11_parede_interna_lod_2
 - c. Casa11_telhado_lod_2
 - d. Casa11_porta_lod_2.
- 3) Utilizar o recurso do 3DS Max de *create shapes line*, para criar as paredes sobre o contorno da planta completa do bairro visível na camada (*layer*) descrita no passo 1).
- 4) Aproveitar o recurso de *geometry* do 3DS Max para modelação dos telhados e portas para cada casa, logo em seguida realizar o *Attach* através do *Modifier list*, pelo recurso de *Editable Poly* do 3DS Max.

A Tabela 15 apresenta a variação do número de polígonos das casas XI, XII e XV em função do nível de detalhe, verificando-se o aumento do número de polígonos à medida que aumenta o detalhe dos modelos. O número de polígono cresce substancialmente no nível de detalhe LOD_2, pois a utilização de telhas com detalhes próximo dos telhados reais aumenta o número de polígonos

Casas	LOD	Número de Polígonos
Casa11-12 e 15	0	305
Casa11-12 e 15	1	800
Casa11-12 e 15	2	56.483

Tabela 15 –Número de polígonos em cada LOD.



Figura 22 – Telhado com LOD 2 da casa XI, com plug-in ATiles.

Na Figura 22 destaca-se o telhado da casa XI, no nível de detalhe dois, desenhado com recurso ao *plug-in* ATiles, onde somente o telhado possui 15.725 polígonos.

A plataforma de exportação do 3D Studio Max, além do formato Autodesk FBX, permite exportação em outros formatos, como o OBJ, utilizado para exportar para plataforma do CityEngine (Figura 23). Para exportação direta para o Unity 3D, obtêm-se melhores resultados com o formato FBX, recorrendo à ferramenta *Level of detail utility*⁴⁸.

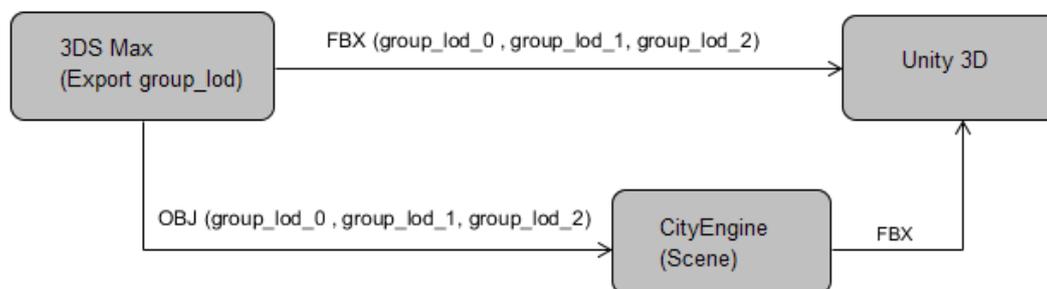


Figura 23 – Processo de exportação de grupo de LOD no 3DS Max.

Níveis de detalhe no CityEngine

O CityEngine não possui um sistema de geração de imagem baseado na proximidade da câmara, por isso não alterna automaticamente entre níveis de detalhe diferentes. Contudo, permite criar diferentes níveis de detalhe para o mesmo objeto e o utilizador pode seleccionar qual a representação que deve estar visível. Para exportar os níveis de detalhe para Unity 3D, é necessário exportar nível a nível.

⁴⁸ 3ds Max, “Level of Detail Utility”, acedido a 11 de junho de 2016, em: <http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/index.html?url=files/GUID-D112D015-8BE6-4172-B816-B5432A50F911.htm,topicNumber=d30e166208>

Os modelos criados no 3DS Max com 3 níveis de detalhe foram importados para o CityEngine separando os diferentes níveis.

No CityEngine modelou-se uma casa com 3 níveis de detalhe. Além da modelação manual, designada no CityEngine por modelação procedimental, combinou-se esta modelação com regras e por isso este modelo está descrito na secção de modelação procedimental

O anexo G apresenta imagens desta casa com os seguintes níveis de detalhe:

- LOD_0: modelo com polígono sem profundidade com textura.
 - LOD_1: adicionou-se ao LOD_0 a textura da porta e da barra.
 - LOD_2: modelos com inserção de *assets*, correspondentes à porta e à barra.
- Neste exemplo explorou-se a capacidade de adicionar objetos geométricos ao modelo para alcançar maiores níveis de detalhe.

4.3.1.2 Aplicação de textura

A exportação do ficheiro do AutoCAD para o ambiente 3DS Max levantou problemas de completude das texturas de alguns objetos 3D, especificamente, o telhado das casas. Por este motivo, foi necessário aplicar de novo as texturas no 3DS Max. Importou-se o ficheiro DXF do AutoCAD para o 3DS Max para aplicar as texturas disponíveis neste ambiente. Uma vez que os modelos foram construídos no AutoCAD com a unidade metros, foi necessário realizar a configuração do ambiente do 3DS Max para esta unidade.

Acrescenta-se ao detalhe de configuração de importação do ficheiro DXF a escolha das camadas (*layers*) associadas à aplicação de texturas. Após importação e definição das camadas, aplicam-se os recursos de texturas. O *modifier list* é um recurso do 3DS Max que permite realizar a modificação de objetos e utilização de modificadores para aplicar diversos recursos com objetivo de alcançar melhor performance da textura do objeto. De entre a diversidade de opções, destacam-se os seguintes no painel *modifier List*:

Edit Poly: Recurso que permite editar objeto de diversas maneiras. O modificador *edit poly* possui como característica a transformação de malhas em polígonos. Utiliza-se a edição por polígono para editar texturas das casas na modelação 3D.

O 3D Studio Max permite trabalhar com material de textura em diferentes níveis de complexidade. No entanto, torna-se fundamental definir os parâmetros básicos no

3DS Max que possuem compatibilidade com o CityEngine e que são os seguintes: i) *Diffuse* – cor refletida pelo objeto quando bem iluminado; ii) *Specular* – cor em que é refletido o foco de luz quando incide em objetos brilhantes; iii) *Ambient* – é a cor refletida pelo objeto quando iluminado pela luz ambiente. Os objetos exportam-se para o formato OBJ, no que resulta os ficheiros OBJ (geometria) e MTL (mapeamento de textura).

Os tipos de sombreadores estabelecem a luz distribuída na superfície. Enfatiza-se o *Phong*, que cria transições suaves de cor entre faces e apresenta realisticamente os pontos de reflexo da luz. O sombreador permite utilizar efeitos de relevo (*bump*), opacidade (*opacity*), brilho, cor especular e mapas de reflexão.

Quanto aos mapas, em (Santos, p. 308, 2013) afirma-se: “Os mapas são texturas que podem ser atribuídos aos materiais. Um material *standard* que contém um ou mais mapas denomina-se material mapeado”. Os materiais mapeados utilizado através da ferramenta *3D Studio Max* possibilitaram atribuir texturas aos materiais, conseqüentemente a substituição do *diffuse* do material. Com a aplicação da textura, pretende-se criar realismo nos objetos. De entre a diversidade de tipos de mapeamento, utiliza-se o mapeamento de relevo *Bump* para provocar relevo para obter rugosidade na textura. Note-se que para atingir este efeito de relevo sobre o material, se atribui ao pixel a cor branca para elevar a superfície e a cor preta para escavar a superfície. Este tipo de textura foi aplicado nas paredes das casas para simular a rugosidade nas paredes.

O ambiente *3D Studio Max* possui como recursos outras técnicas, por exemplo *UVW Map*: recurso que permite alterar as coordenadas de X, Y e Z das texturas dos objetos. O “x”, “y” e “z” correspondem respetivamente ao “U”, “V” e “W”. Torna-se possível adicionar imagem através de *Bitmap* e realizar manipulação de escala, rotação e movimento da imagem para adequação precisa no objeto a ser texturizado.

O recurso de *Smoothing Group* possibilita aplicar suavização em polígonos. Este recurso utiliza-se com a edição de polígono do tipo *Auto smooth*, que permite aplicar suavização, de acordo com o ângulo entre as faces. Por conseguinte, na textura dos objetos modelados torna-se proveitoso o uso de *UVW Map*, principalmente para os telhados da casa para alcançar o posicionamento adequado da textura na geometria do telhado, cuja característica principal consiste em ser inclinada para o pátio interior.

Texturas no CityEngine

O tratamento de materiais no CityEngine baseia-se nos atributos de *shape*, relativos a sombreamento, texturização e exportação da geometria das formas. Existem seis canais de textura com uma semântica fixa descrita na Tabela 16

Attribute	Description
string material.name	O nome usado para referenciar materiais nos arquivos de modelo de exportação.
string material.shader	Utilizado para exportar pelo CityEngine.
float material.color.{r g b}	<i>Diffuse color</i> . Parâmetro individual de componente de cor, possui como padrão branco.
string material.color.rgb	<i>Diffuse color</i> . Parâmetro para componente cor através do formato hexadecimal. Exemplo para vermelho: "#FF0000".
float material.ambient.{r g b}	<i>Ambient color</i> . Parâmetro individual para cada componente, por padrão é preto.
float material.specular.{r g b}	<i>Specular color</i> . Parâmetro Individual para cada componente de cor, por padrão é preto.
float material.opacity	<i>Opacity factor</i> . Valores de 1 a 0: valor de um, possui opacidade completa; valores de zero, possui transparência.
float material.reflectivity	<i>Reflectivity factor</i> . Valores entre zero (não reflexão) a um (possui reflexão): valores em zero, não reflexão; valores em um, possui reflexão. Nota: a reflexão depende de <i>Specular Color</i> .

Tabela 16 – Parâmetros de textura no CityEngine adaptado⁴⁹.

Os objetos com as texturas criadas no 3DS Max podem ser exportados para o CityEngine através do formato OBJ, que permite a compatibilidade de modelos tridimensionais sólidos ou com camada de textura única vindo de qualquer ferramenta de *Digital Content Creation Tools* – DCC.

O *3D Studio Max* possui os seguintes ficheiros de exportação para o formato *Wavefront OBJ*:

- *File name.obj*: geometria do objeto.
- *File name.mtl*: parâmetros de configuração do material do objeto.
- *Diretoria maps*: mapeamento e localização das texturas referenciado pelo ficheiro *mtl*.

A Tabela 17, apresenta dois ficheiros MTL gerados pela exportação no formato OBJ: à esquerda pelo *3D Studio Max* e à direita pelo *CityEngine*.

⁴⁹ CityEngine, “Material shape Attributes”, acessado a 29 de junho de 2016, em: http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/cgareference/op_texture.html

3DS Max	CityEngine
<pre># 3ds Max Wavefront OBJ Exporter v0.97b - (c)2007 guruware # File Created: 12.02.2016 13:39:44 newmtl 02___Telhado Ns 10.0000 Ni 1.5000 d 1.0000 Tr 1.0000 Tf 1.0000 1.0000 1.0000 illum 2 Ka 0.5882 0.5882 0.5882 Kd 0.5882 0.5882 0.5882 Ks 0.0000 0.0000 0.0000 Ke 0.0000 0.0000 0.0000 map_Ka maps/Roof.Tiles.jpg map_Kd maps/Roof.Tiles.jpg</pre>	<pre>newmtl CityEngineShapeMaterial_1 illum 3 Ka 1 1 1 Kd 1 1 1 Ks 0 0 0 Ns 0 d 1 Tf 1.0 1.0 1.0 Ni 1.0</pre>

Tabela 17 – Ficheiro mtl gerados no 3DS Max e no CityEngine.

Os parâmetros *ka* (*ambiente reflexion*), *kd* (*diffuse reflexion*) e *ks* (*specular reflexion*) variam entre zero (0) e um (1).

Na coluna esquerda, correspondente ao ficheiro gerado pelo 3DS Max, o valor para *illum* é dois, o que significa que é aplicada a relexão especular do tipo *Phong* na textura. Na coluna direita, correspondente ao ficheiro gerado pelo CityEngine, evidencia-se o parâmetro *illum* com valor três, o que corresponde a aplicar o operador *projectUV(0)* na cor do material Os parâmetros *Tf* e *Ni* garantem, neste exemplo, a compatibilidade com a ferramenta de modelação *Maya*, mantendo a interoperabilidade de dados tridimensional entre as ferramentas DCC.

4.3.2 Modelação procedimental

O CityEngine permite o desenvolvimento de ambientes urbanos virtuais de maneira automatizada através de *shape grammar rule files*. Estes ficheiros de regras consistem numa linguagem de programação exclusiva para gerar conteúdo tridimensional arquitetónico (Haegler et al. 2009). Para alcançar dinamismo e eficiência na construção de ambientes virtuais tridimensionais, utiliza-se o ambiente de desenvolvimento do CityEngine, designado por *Computer Generated Architecture* (CGA), que permite a criação e aplicação das regras para criação dos modelos tridimensionais.

Além do desenvolvimento de modelos através de regras, o CityEngine oferece também o recurso de modelação poligonal (*polygonal modeling*) para criação de modelos tridimensionais sobre *shapes*. Estes modelos podem ser enriquecidos combinando a modelação poligonal com a aplicação de regras, correspondendo assim a uma forma alternativa de aplicação de modelação procedimental.

A modelação procedimental caracteriza-se pela criação de ficheiros de regras definidas recorrendo aos operadores oferecidos pela *CGA shape grammar*. As *shapes*, que consistem em uma geometria e em uma caixa delimitadora (*bounding box*), chamada âmbito (*scope*), são o conceito central para aplicação de regras. Existe uma regra inicial, designada por *@Start Rule*, que é a primeira a ser executada sobre a(s) *shape(s)* selecionada(s) e que desencadeia a aplicação das regras seguintes. A aplicação de uma regra gera uma (ou eventualmente mais do que uma) *shape*, cujo nome, designado por *shape symbol*, é simultaneamente o nome da regra que é aplicada a esta *shape* para gerar uma *shape* sucessora.

Esta aplicação sucessiva de regras cria uma hierarquia de *shapes* e apenas as folhas desta árvore constituem o modelo final. A Figura 24, à esquerda, apresenta o âmbito de uma *shape* dentro de uma caixa delimitadora e, à direita, a árvore de *shapes* que resultou da aplicação das regras para a sua geração.

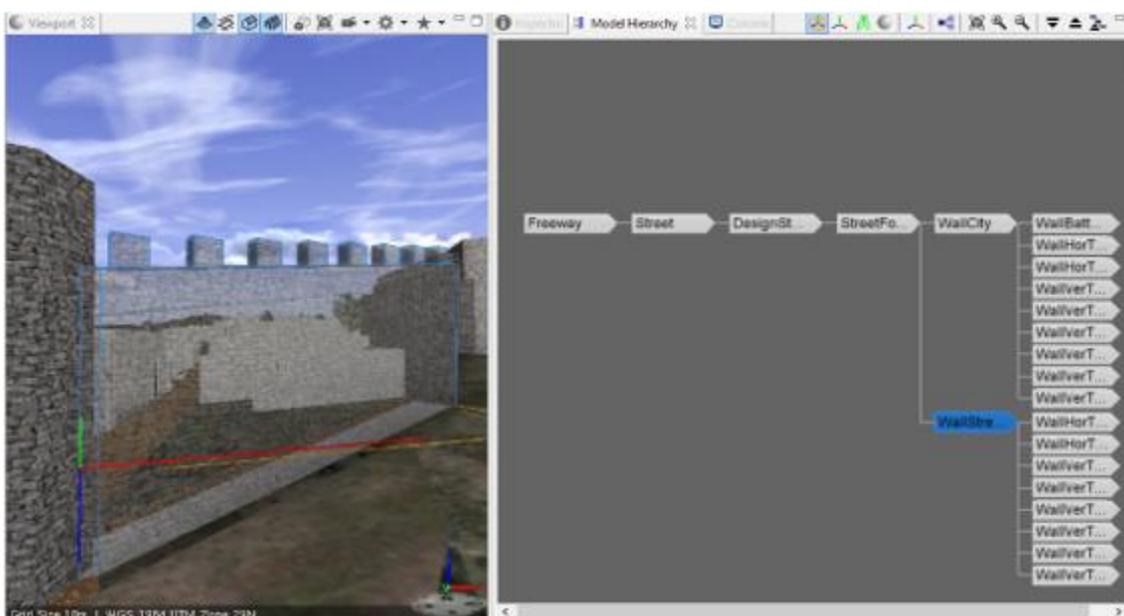


Figura 24 – Geometria dentro do âmbito e a respetiva árvore de shapes.

Em seguida descrevem-se as estruturas construídas com cada das técnicas: modelação procedimental – colocação de vegetação no cenário, aplicação de textura nas

ruas, construção do modelo 3D da muralha e casas do tipo açoteia; modelação poligonal com código de regras - casas com telhados inclinado.

4.3.2.1 Estruturas construídas com modelação procedimental

Descrevem-se em seguida os elementos do cenário de Mértola Virtual construídos recorrendo à modelação procedimental.

Colocação de vegetação

Para a colocação de vegetação no ambiente urbano recorreu-se a uma biblioteca de vegetação⁵⁰ disponibilizada pela ESRI para o ambiente CityEngine. A *Vegetation library*⁵¹ possui modelos de árvores 3D em três diferentes níveis de detalhes: *Model, Fan and Analytical*. Os modelos de árvore 3D disponibilizados na biblioteca de vegetação foram construídos utilizando *The Plant Factory*⁵², segundo a equipa da ESRI.

O ficheiro contém as regras para criar a paisagem usando uma textura para o chão e adicionando alguns tipos de árvore. Das árvores disponíveis na biblioteca escolheram-se as que são mais adequadas à paisagem de Mértola de modo a compor um cenário realista com vegetação 3D. Para este efeito teve-se em atenção a descrição de Gomes (2002) quanto ao coberto vegetal da região de Mértola no século XI e XII.

O ficheiro *landscape.cga* possui duas regras determinantes para distribuição de vegetação: *LotLandscape* e *LotTree*. A regra *LotLandscape* é responsável por aplicar às *shapes* que constituem o chão uma textura com um padrão de vegetação. A imagem usada para esta textura foi retirada de uma imagem do Google Earth numa área da região de Mértola. A regra *LotTree* distribui a vegetação de árvores 3D sobre uma determinada área a partir da parametrização de *block parameters*⁵³.

Apresenta-se em seguida um extrato do código de regras do ficheiro *landscape.cga*, relativo às regras *LotTree* e *LotLandscape*.

```
@StartRule
LotTree -->
    alignScopeToAxes (y)
    LotAligned
```

⁵⁰ <https://www.arcgis.com/home/item.html?id=766310c898db4b77ad1d73262433a1be>

⁵¹ ArcGIS, “Vegetation Library with LumenRT 3D Plants”, acessado a 28 de junho de 2016, em: <https://www.arcgis.com/home/item.html?id=766310c898db4b77ad1d73262433a1be>

⁵² <http://www.plantfactory-tech.com/>

⁵³ CityEngine, “Block Parameters”, acessado a 05 de julho de 2016, em: http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/is/create/block_params.html

```

LotAligned -->
  case scope.sx < 2 && scope.sz < 2 && geometry.area > 45:
    GreenCity
    setback(1.5) { front : GreenCity | remainder : NIL }
  else:
    GreenCity
    GreenTreesCity
    setback(14.5) { front : GreenCity | remainder : NIL }

@StartRule
LotLandscape -->
  GreenCity

GreenCity -->
  alignScopeToAxes (y)
  setupProjection(0, scope.xz, projectionWidth, projectionWidth,0)
  projectUV(0)
  texture(TextureLandscape)
  translateUV(0, rand(1), rand(1))
  rotateUV(0, rand(360))
  Field.

```

O extrato de código seguinte é responsável por chamar a biblioteca de vegetação. Note-se que a lista com o tipo de árvores, a colocar na paisagem, contém as espécies mais próximas das que existem na região de Mértola.

```

// Sets parameters for tree - view
getTree = listRandom("Ficus;Pawpaw;Smooth Sumac")

// Create tree
// Call patterns tree - file Plant_Loader.cga
// Biblioteca que possui as definições de árvores numa lista aleatória
CreateTree -->
  Plant.Generate(getTree)

```

A Figura 25 apresenta, à esquerda, as regras para a colocação de vegetação, ou seja, tem como *Start Rules* as regras LotLandscape ou LotTree; e, à direita, a imagem do modelo 3D resultante da aplicação da regra LotTree numa *shape* em destaque.



Figura 25 – Regras para distribuição de vegetação 3D.

Aplicação de textura às ruas

Para obter mais realismo, adicionou texturas às ruas de modo a simular um chão de terra batida de acordo com a época do período islâmico, segundo informações arqueológicas (Martínez et al., 2009). Para este efeito criou-se o ficheiro de regras RuleStreet.cga.

Para a textura acompanhar as ruas previamente definidas, os parâmetros para o operador *offset* têm em atenção a parte interior da *shape* que vai receber a textura. Para este efeito, recorre-se à função *getGeoCoord()* que seleciona a coordenada do sistema utilizada na cena (WGS 1984 UTM zone 29N). Segue-se um extracto deste código.

```
// Sets for offset operation of texture interna
// Função getGeoCoord() % size relaciona-se a coordenada de sistema
textureOffsetX(size) = getGeoCoord(X) % size
textureOffsetZ(size) = -getGeoCoord(Y) % size
```

O extrato de código seguinte contém a regra inicial MenuStree para gerar a aplicação de texturas nas ruas.

```
@StartRule
MenuStreet -->
    StreetTexturing

// Rule for texturing
StreetTexturing -->
    alignScopeToAxes()
    // offset polygons are constructed in the interior (negative
    //sign) or in the exterior (positive sign), respectively.
    setupProjection(0,scope.xz,textureDim,textureDim, -
    textureOffsetX(textureDim), -textureOffsetZ(textureDim))
    projectUV(0)
    texture(street_texture)
    setupProjection(2,scope.xz,streetDirtDim,streetDirtDim, -
    textureOffsetX(streetDirtDim), -textureOffsetZ(streetDirtDim))
    projectUV(2)
    // layer 2 with dirtmap for pavement
    //set(material.dirtmap, street_texture)
    Street.
```

No extrato de código acima, a regra StreetTexturing possui o parâmetro *set* com a camada dois (*layer 2*) com *dirtmap* com comentários. Esta camada poderá ser usada no caso de se pretender escurecer mais a textura de terra batida, resultando na sobreposição da textura. A identificação da camada encontra-se na Tabela 18 onde se descrevem os seis canais de texturas que o CityEngine possui e os respectivos valores UV. Através do operador “set” faz-se a ligação entre os canais de textura e as componentes a que se aplicam para obter diferentes efeitos.

UV - Definição	Camada de Textura
UV-Set=0	TextureLayer= colormap
UV-Set=1	TextureLayer= Bumpmap
UV-Set=2	TextureLayer= Dirtmap
UV-Set=3	TextureLayer= Specularmap
UV-Set=4	TextureLayer= Opacitymap
UV-Set=5	TextureLayer= Normalmap

Tabela 18 – Canais de textura no CityEngine adaptado.

Construção da muralha

A construção da muralha recorreu ao procedimento de construção de ruas do CityEngine (*grow street*). Através deste recurso, definiram-se ruas para servir de base ao levantamento da muralha. Além disso, utilizaram-se modelos de torres construídas por modelação poligonal para adicionar em determinadas posições da muralha. O conjunto de regras para geração da muralha está descrito no ficheiro RuleWallCity.cga.

Após a definição de ruas através do recurso *grow street*, houve um ajustamento manual do traçado das ruas para aproximar a base da muralha, conforme o desenho das muralhas em Martínez (2009), Macias (1996b) e Armas (1509) e as imagens atuais obtidas pelo Google Earth.

Posteriormente, definiram-se os nós como parâmetros de intersecção de ruas e que seviram para adicionar as torres e realizar a junção com as restantes partes da muralha.

A Figura 26, à esquerda, realça um nó resultante de intersecção de ruas, onde foi colocada uma torre, e a imagem da direita ilustra o resultado da aplicação da regra para adição de uma torre à muralha.

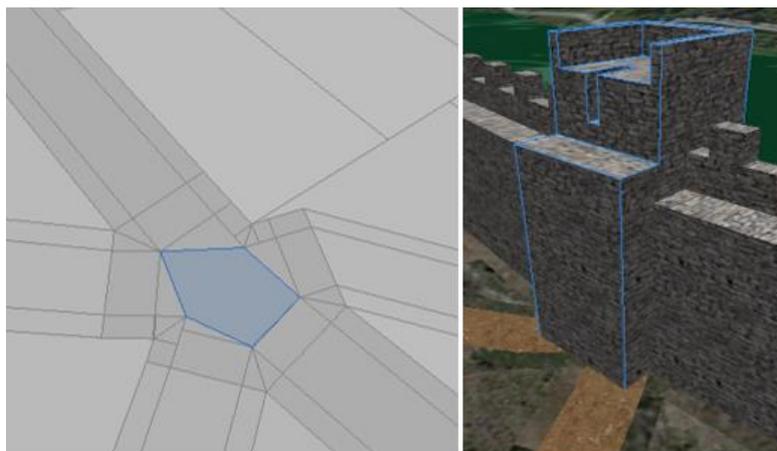


Figura 26 – Inserção de torre na cena com código de regra.

Como aspeto positivo realça-se a flexibilidade de adicionar ou reduzir o número de torres e a variabilidade de objetos (*assets*) na cena CityEngine através da aplicação da regra para sobrepôr torres às *shapes streets* que definem a base da muralha.

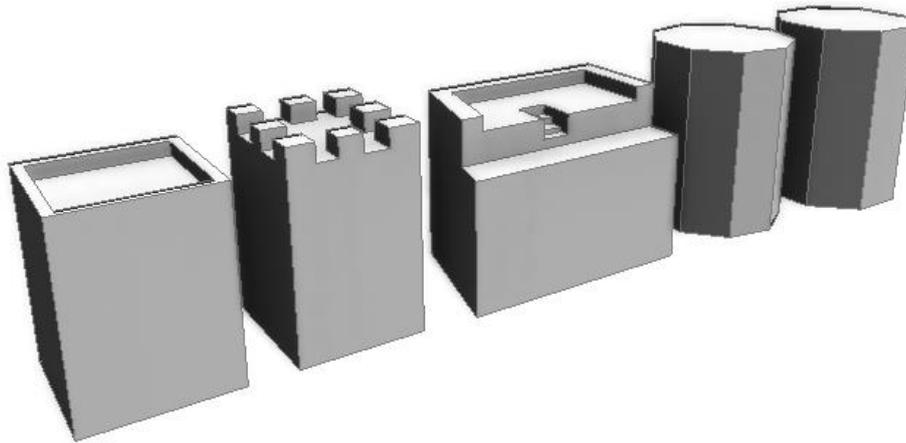


Figura 27 – Modelos geométricos para exportação.

A Figura 27 apresenta modelos criados no ambiente CityEngine recorrendo a modelação poligonal, que são exportados para o formato OBJ para serem utilizados como ativos (*assets*) pelo código de regras. Como já referido, a exportação para o formato OBJ gera um ficheiro com a geometria (OBJ) e outro com o mapeamento da textura (MTL) numa determinada diretoria do projeto.

A Figura 28 mostra, à esquerda, em cima, a seleção de um ficheiro (formato OBJ) que descreve uma torre e, em baixo, o modelo que lhe corresponde. Nesta figura, à direita, ilustra-se o resultado da inserção do modelo da torre na cena após a execução do código de regras de geração do modelo 3D da muralha.

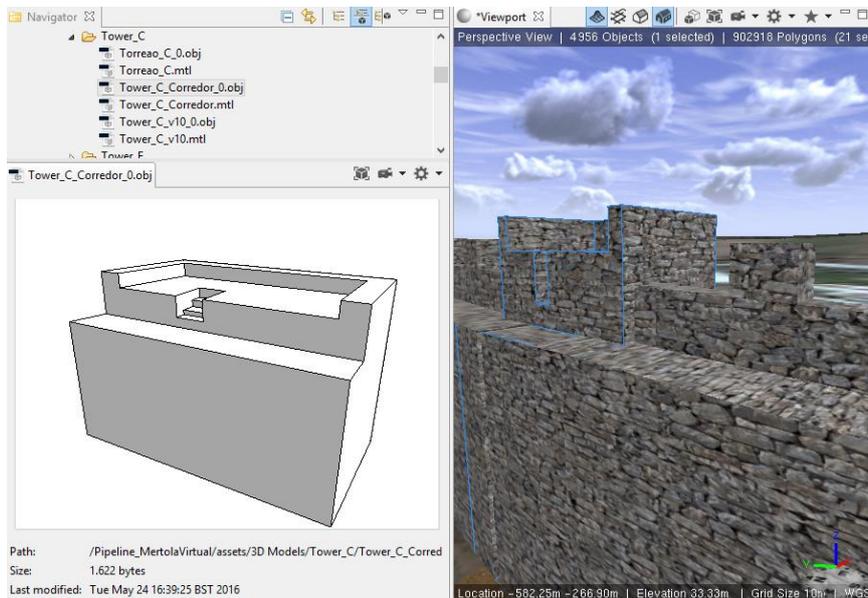


Figura 28 – Modelo de uma torre e o resultado da sua inserção no modelo 3D.

O extrato de código seguinte contém a regra (`InsertTower`) para adicionar uma torre à muralha. O argumento do operador “`i`” corresponde ao caminho para chegar ao

ficheiro com o modelo da torre, representado sem textura, à esquerda, na Figura 28. A regra TowerSplit é responsável pela aplicação de textura para obter o modelo representado na figura à direita.

```

InsertTower -->
  s('2,towerRiverHeight,'1.9)
  r(0,towerAngle,0)
  center(xz)
  i("3D Models/Tower_C/Tower_C_Corredor_0.obj")
  t(0,-towerEarthDepth,-1.5)
  TowerSplit(scope.sz)

TowerSplit(depth) -->
  comp(f) {
    top: WallHorTexturing
    | 0: WallVerTexturing
    | 4: WallVerTexturing
    | 2: WallVerTexturing
    | 2: WallVerTexturing
    | all: WallVerTexturing}
// end tower in scene

```

A principal característica da aplicação de textura consiste em dividir as faces da geometria através do operador $comp(f)$, após a regra predecessora TowerSplit, e aplicar a textura com os parâmetros definidos com a regra WallHorTexturing e WallVerTexturing – texturas horizontal e vertical, respetivamente.

A largura da muralha depende da largura que foi definida para a rua que lhe serve de base. Assim sendo, as regras para a construção do modelo da muralha vão usar os valores previamente estabelecidos para este parâmetro, já referido na Tabela 15 acerca da construção das ruas.

Na Figura 29 podem comparar-se as *shapes* antes (lado esquerdo) e após (lado direito) a aplicação do código de regras para geração do modelo 3D da muralha.

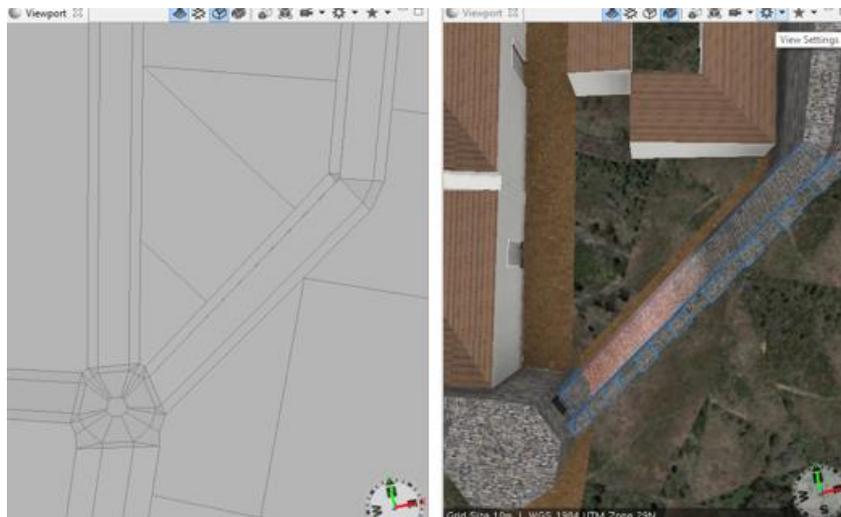


Figura 29 – Comparação entre street shape e modelo 3D da muralha.

O extrato do código de regras abaixo lista as *shapes* predecessoras que geram as *shapes* sucessoras para criar o modelo 3D da muralha.

```
@Group("Generare Models of WallCity")
// wall
Freeway      -->  Street
Joint        -->  Street
Junction     -->  Street
```

O nome destas regras predecessoras corresponde aos diferentes tipos de *shapes* que a interseção de ruas do CityEngine⁵⁴ pode gerar (*Freeway*, *Joint*, *Junction*, entre outras).

Na Figura 30 destacam-se as etapas para geração do modelo 3D: no painel à esquerda, seleciona-se o ficheiro de código de regras RuleWallCity.cga; no painel central, listam-se as opções de regras iniciais e encontra-se selecionada a regra Freeway; no painel à direita, mostra-se o modelo tridimensional de parte da muralha que resulta de aplicar a regra Freeway à *shape* (ou *shapes*) que estiverem selecionadas.

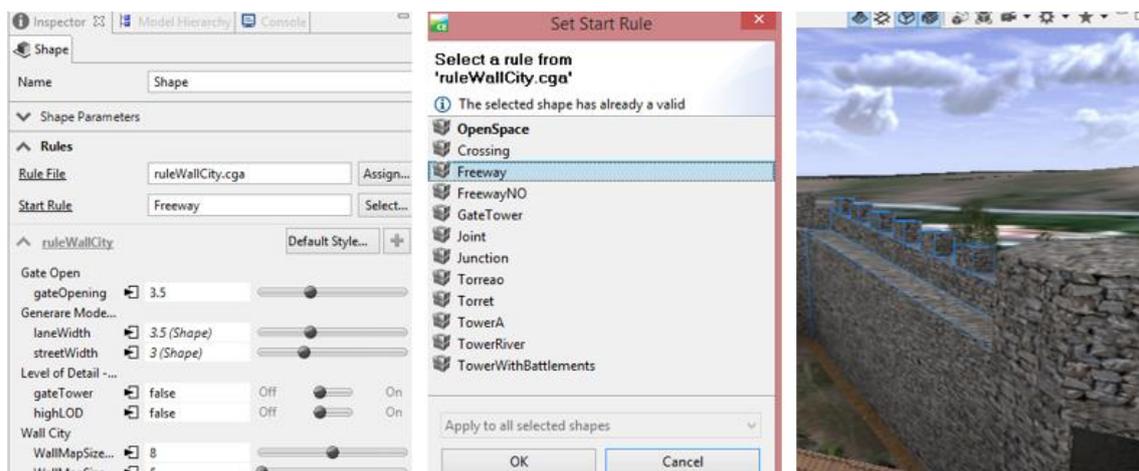


Figura 30 – Etapas de geração do modelo 3D da muralha.

A construção completa da muralha resulta de selecionar sequencialmente as *shapes* que servem de base às várias componentes da muralha (muralha com e sem ameia e vários tipos de torres) e aplicar a respetiva regra. A escolha do tipo de torre a colocar em cada nó teve como base os desenhos de Armas (1509), Macias (1996b) e Martínez (2009).

⁵⁴ CityEngine, “Street and Crossing Shapes”, acessado a 07 de junho de 2016, em: http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/is/create/streets_hapes.html

Construção das casas tipo açoteia

Para permitir uma maior diversidade de casas, construiu-se um modelo de casa baseado em modelação procedimental. Neste modelo experimentam-se as várias opções do CityEngine para gerar os telhados, com o objetivo de obter telhados inclinados para o pátio interior, como requerido na modelação das casas do período Islâmico. Contudo, não se atingiu este objetivo apenas com a utilização de regras e foi necessário aplicar a técnica de modelação poligonal combinada com regras, como se explica noutra secção.

O ficheiro RuleGenerateHouse.cga contém as regras para a modelação de uma casa a partir de uma *shape* de dimensão e forma variáveis. Neste código investigamos a variação do tipo do telhado que pode ter os operadores padrões *roofFlat*, *roofGable*, *roofHip* e *roofShed* como apresentado na Figura 31 (a), (b), (c) e (d), respetivamente.

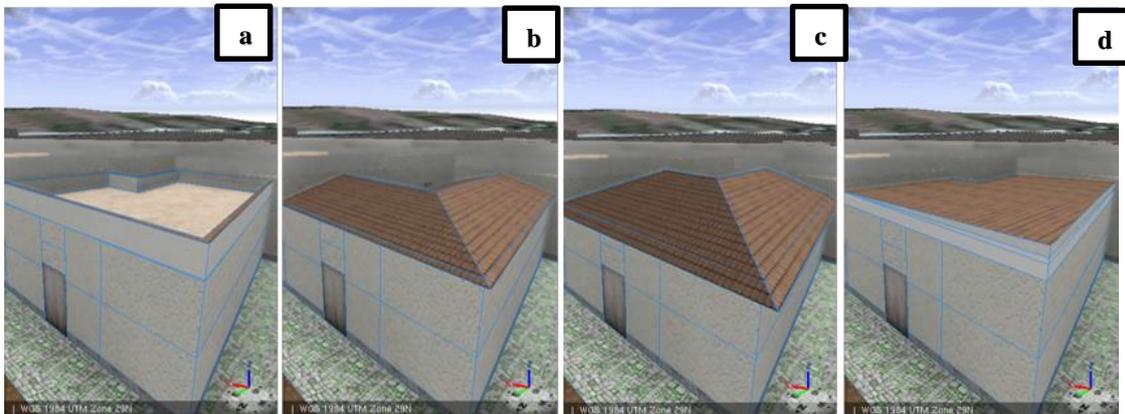


Figura 31 – Variadade de tipo de telhado com operadores padrões.

O extrato do código de regra abaixo define os parâmetros dos vários tipos de telhados:

```
// roof texture is applied
Roof -->
    //color(wallColor)
    //setupProjection(0, scope.xy, scope.sx, scope.sy)
    //texture(roof_tex)
    //projectUV(0)
    case Rooftype == "Gable" : RoofGable
    case Rooftype == "Hip"   : HipRoof
    case Rooftype == "Shed"  : ShedRoof
    case Rooftype == "Flat"  : FlatRoof
    else: FinalShape.
```

A Figura 31 (a) corresponde à casa do tipo açoteia (casas algarvias), cujo o telhado é modelado com o extrato de código abaixo. Neste código, a regra FlatRoof, que é invocada quando o tipo de telhado for igual a Flat, tem o operador *offset* com um valor negativo, com o objetivo de interiorizar a cobertura para simular um parapeito.

```
FlatRoof -->
    Texture("Roof_house_acoteia") offset(-0.2, border) extrude(0.8)
```

```
comp(f){ top : Texture("Roofhouse") | side : Texture
("Roof_house_wall")
}
```

Texturas aplicadas na modelação com regras

A aplicação de textura por meio de regras possui como objetivo automatizar a sua adição ao modelo 3D. Este modo de aplicação de textura foi usado no chão dos pátios interiores, nas paredes, nos telhados, nas portas, nas ruas, no solo e nas muralhas. No anexo D apresentam-se duas tabelas com as imagens das texturas e um pequeno extrato do código de regras relativo à sua aplicação.

Para o processamento do modelo 3D não sobrecarregar a memória e placa de vídeo local da máquina, as texturas catalogadas nas tabelas utilizaram, sempre que possível, as recomendações do CityEngine⁵⁵ para o melhor desempenho: tamanho da textura de 500 x 500 *pixels* e 8 bit de profundidade.

O CityEngine carrega o modelo 3D juntamente com a textura para visualização diretamente na placa de vídeo dedicado VRAM. Para um ótimo desempenho de carregamento do modelo 3D com textura, recomenda-se 1 GB dedicado para VRAM (placa gráfica com sua própria memória).

4.3.2.2 Estruturas construídas com modelação poligonal combinada com regras

A técnica de modelação poligonal, que permite criar modelos tridimensionais sobre *shapes*, pode ser combinada com código de regras⁵⁶. A aplicação de regras consiste em substituir polígonos do modelo original por outros modelos. Estes modelos podem ser criados por modelação poligonal, em seguida aplica-se a opção de exportação para o formato *Wavefront OBJ*, criando um novo modelo (*asset*) que pode ser invocado na aplicação de regras.

No cenário de Mértola Virtual a técnica de modelação poligonal com código de regras foi aplicada na criação de casas com telhado inclinado para o pátio interior.

⁵⁵ GeoNet, “The Esri Community”, acessado a 06 de junho de 2016, em: <https://geonet.esri.com/thread/48433>

⁵⁶ ESRI CityEngine, “Tutorial 14c: Combining polygonal modeling with rules”, acessado a 06 de junho de 2016, em: <http://desktop.arcgis.com/en/cityengine/latest/tutorials/tutorial-14c-combining-polygonal-modeling-with-rules.htm>

As casas foram criadas por modelação poligonal e aplicaram-se texturas. Posteriormente, aplicou-se um conjunto de regras que substituiu algumas texturas por objetos para proporcionar maiores detalhes. Os parâmetros destes objetos podem ser ajustados para se adaptarem melhor ao modelo onde vão ser colocados.

A Figura 32 mostra o modelo de uma casa onde se inseriu uma porta através de uma regra. No ficheiro de regras são definidos os valores dos atributos altura e largura da porta, construída previamente, bem como, os parâmetros de altura e largura para as texturas que se mantêm (parede, telhado, pátio interior e barra). Os valores dos atributos foram definidos com seguinte extrato de código:

```
version "2014.0"
// Attributes Doors
attr doorWidth = 1
attr doorHeight = 2

// Wall
attr wallWidth = 1
attr wallHeight = 2

// Roof
attr roofWidth = 1
attr roofHeight = 2

// Courtyard
attr courtyardWidth = 2
attr courtyardHeight = 2

// Foundation
attr foundationWidth = 10
attr foundationHeight = 2
```

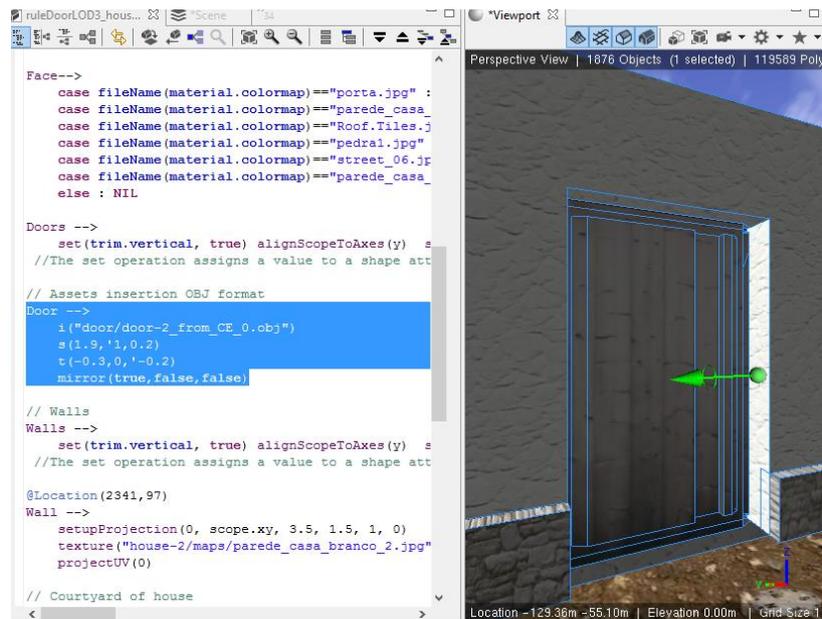


Figura 32 – Modelação poligonal com código de regra.

A Figura 33, tem na imagem do lado esquerdo, a porta inserida no modelo ativada, o que permite visualizar, na imagem do lado direito, a árvore de *shapes* de todo o modelo, evidenciando a porta ativada. Após construção do novo modelo da casa, este pode ser exportado para o formato OBJ para ser adicionado a um ambiente urbano virtual no CityEngine.

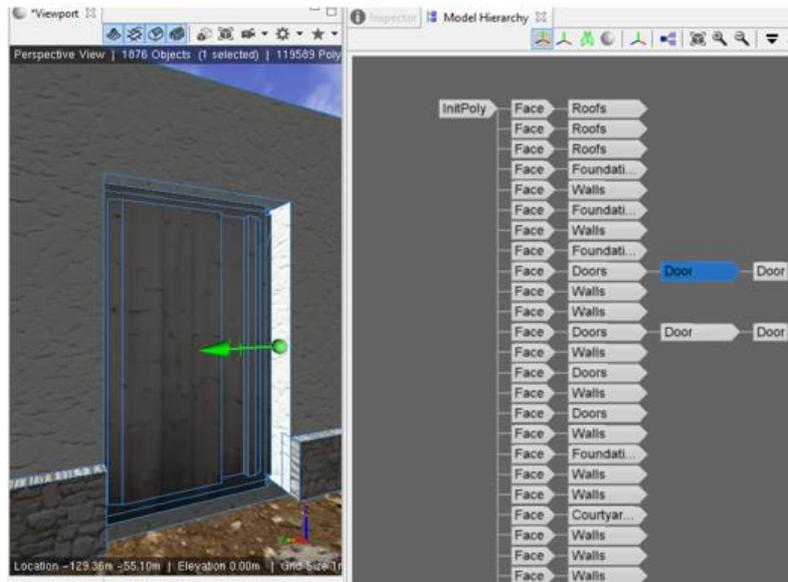


Figura 33 – Modelação poligonal com árvore de shapes.

O extrato de código seguinte contém as regras para substituição da textura original da porta (“porta.jpg”) por um objeto através de um operador de inserção, que contém a diretoria do objeto (`i("door/door-2_from_CE_0.obj")`). Neste extrato de código, a regra *InitPoly* é a *predecessor shape* e prossegue com as demais regras – *Face*, *Walls*, *Foundations* e *Door*, que são as sucessoras *shape*.

```
//Esta regra substitui todos os polígonos com texturas door com a door
3D.
@StartRule
InitPoly-->
    // all - selects all components
    comp(f) { all : Face }

Face-->
    case fileName(material.colormap)=="porta.jpg" : Doors
    case fileName(material.colormap)=="parede_casa_branco_2.jpg" :
Walls
    case fileName(material.colormap)=="Roof.Tile
s.jpg" : Roofs
    case fileName(material.colormap)=="pedral.jpg" : Courtyard
    case fileName(material.colormap)=="street_06.jpg" : Courtyard
    case fileName(material.colormap)=="parede_casa_branco_1.jpg" :
Foundations
    else : NIL
```

```

Doors -->
  set(trim.vertical, true) alignScopeToAxes(y)
  split(y){~doorHeight:split(x){~doorWidth: Door})*

// Assets insertion OBJ format
Door -->
  i("door/door-2_from_CE_0.obj")
  s(1.9,'1,0.2)
  t(-0.3,0,'-0.2)
  mirror(true,false,false)

```

A Figura 34 mostra, à esquerda, o modelo da casa construída com a aplicação de regras e, à direita, um extrato do ficheiro MTL que resulta da exportação do modelo para o formato OBJ. Note-se o mapa de textura referenciado no ficheiro MTL (Roof.Tiles.jpg) relativo à textura do telhado.

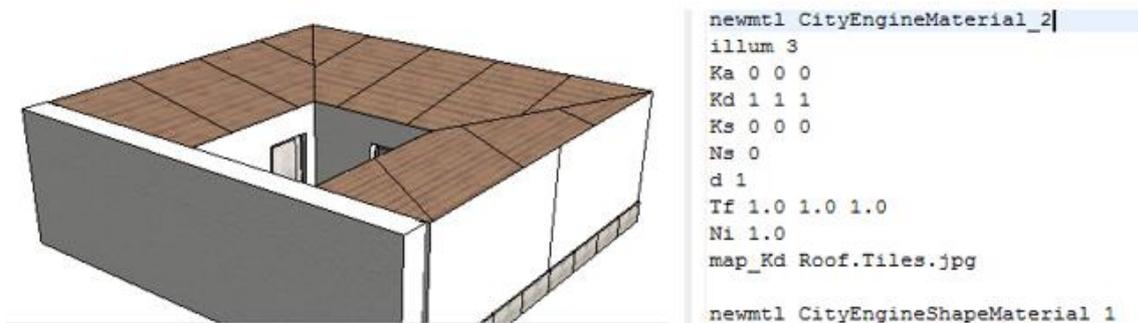


Figura 34 – Modelo da casa gerada e extrato do ficheiro.

4.4 Integração: desenvolvimento do ambiente virtual

Nesta secção resumem-se as principais etapas do desenvolvimento do ambiente urbano virtual. O CityEngine caracteriza-se por ter um fluxo de atividades com etapas sequenciais executadas por diferentes componentes.

A Figura 35 contém as etapas de desenvolvimento para produção do modelo tridimensional do caso de estudo Mértola Virtual. Identificam-se os ficheiros de entrada: imagem do modelo digital do terreno (*Image DEM*), imagem georreferenciada de textura (*Image GEO*) e imagem de obstáculo (*Image Obstacle*). A primeira etapa consiste na criação da cena (*CreateScene*). Com a cena criada, gera-se o terreno (*CreateTerrain*) e, posteriormente, utiliza-se a componente de colocação automatizada de ruas (*Grow Street*) que sobrepõe as *shapes* ao terreno. A integração de modelos tridimensionais pode ser feita através do módulo para codificação de regras (*Createfilerule*) ou por importação (*CoreImportObject*) dos modelos tridimensionais desenvolvidos com ferramentas DCC.

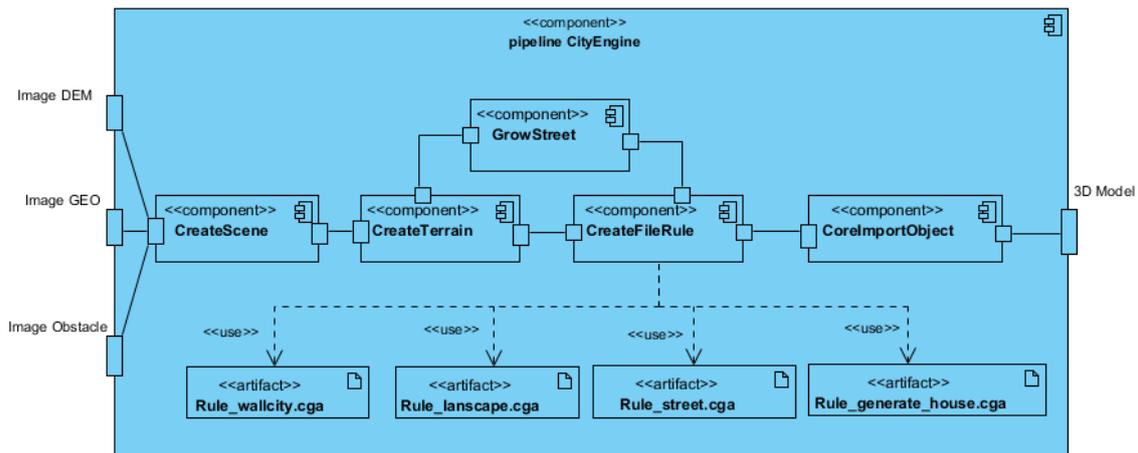


Figura 35 – Pipeline CityEngine.

Apresentam-se em seguida mais detalhes sobre estas etapas:

CreateScene: na criação da cena utilizou-se o sistema de coordenadas WGS 1984 UTM zone 29N, pois este possui compatibilidade com a fonte de dados do modelo digital do terreno adquirido do sítio *Open Topography* e esta de acordo com a área de interesse do caso de estudo.

CreateTerrain: a geração do terreno realizou-se através da criação de uma nova *map layer*, descrita na secção 4.1, que requer um heightmap (*Image DEM*) e uma imagem de textura (*Image GEO*).

GrowStreet: para desenhar as ruas do ambiente urbano virtual foi usado um módulo de crescimento automatizado de ruas, descrito a secção 4.2. Este constrói uma camada de *shapes* sobrepostas ao terreno, tendo por base o heightmap (*Image DEM*) e o ficheiro obstáculo (*Image Obstacle*).

CreateFileRule: para criar alguns dos modelos tridimensionais e também para aplicação de algumas texturas foram criados ficheiros de regras com o editor CGA. A secção 4.3.2 esclarece com mais detalhes o funcionamento da codificação de regras e sua aplicação em *shapes*. Na Figura 35 identificam-se os principais ficheiros de regras usados neste protótipo: *Rule_wallcity.cga*, para criar a muralha; *Rule_landscape.cga*, para colcar a vegetação na paisagem; *Rule_street.cga*, para aplicar a textura nas ruas e *Rule_generate_house.cga*, para a modelação de casas do tipo açoteia.

CoreImportObject: os modelos tridimensionais desenvolvidos por ferramentas de modelação podem ser importados para o CityEngine no formato Wavefront OBJ de duas formas: i) como modelos estáticos (*static models*): não permitem a aplicação de regras, apenas se podem manipular através de rotações, mudanças de escala e

translações (deslocamentos); ii) como modelos em *shape*: ficam compatíveis com as manipulações do CityEngine, isto é, é possível aceder aos seus atributos, *assets* e texturas e, principalmente, aplicar conjuntos de regras. Caso um modelo for importado como estático, é possível aplicar dentro do ambiente uma operação para converter o modelo para *shape model* através da operação *convert model to shape*.

A Figura 36 apresenta o processo de tratamento dos modelos estáticos. A sua importação requer um ficheiro.obj com a geometria e um ficheiro MTL com a textura.

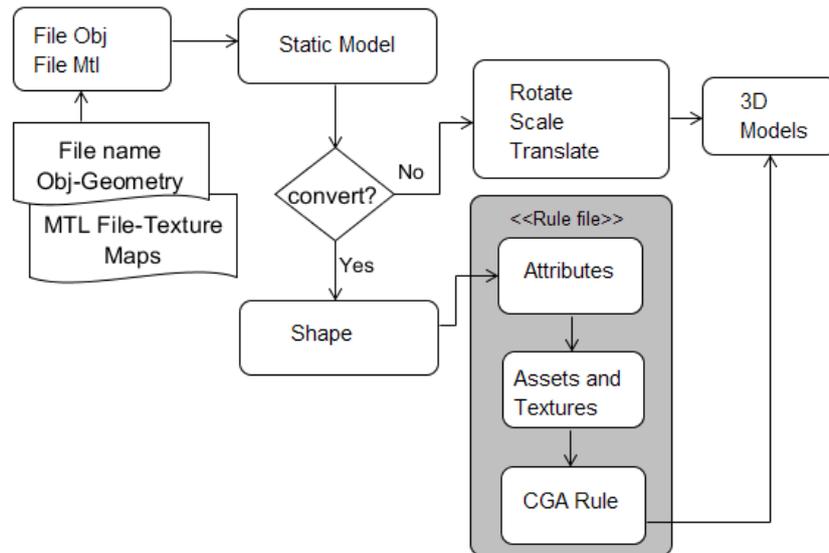


Figura 36 – Processo de importação do modelo estático.

Neste projeto utilizaram-se modelos construídos com os softwares de modelação AutoCAD, Blender e 3DS Max. Os modelos desenhados com o AutoCAD e o Blender foram exportados para o 3DS Max para aplicação de texturas, por isso, todos os modelos foram importados para o CityEngine no formato OBJ, sendo exportado pelo 3DS Max.

Foi necessário alinhar os modelos importados com o terreno. Além das funções de alinhamento *align graph to terrain* e *align the terrain to shapes*, já mencionadas a propósito da geração de terreno, existe outra função de alinhamento: *align static models*. Esta função de alinhamento, foi utilizada para ajustar os modelos estáticos ao terreno.

A Figura 37 apresenta, à direita, a interface de alinhamento para modelos estáticos, em que os parâmetros de alinhamento em destaque são o ficheiro *heightmap* e o *offset* (terreno a ser alinhado e deslocamento para adicionar após o alinhamento à coordenada y, respetivamente). A imagem à esquerda da Figura 37 mostra o resultado do alinhamento.

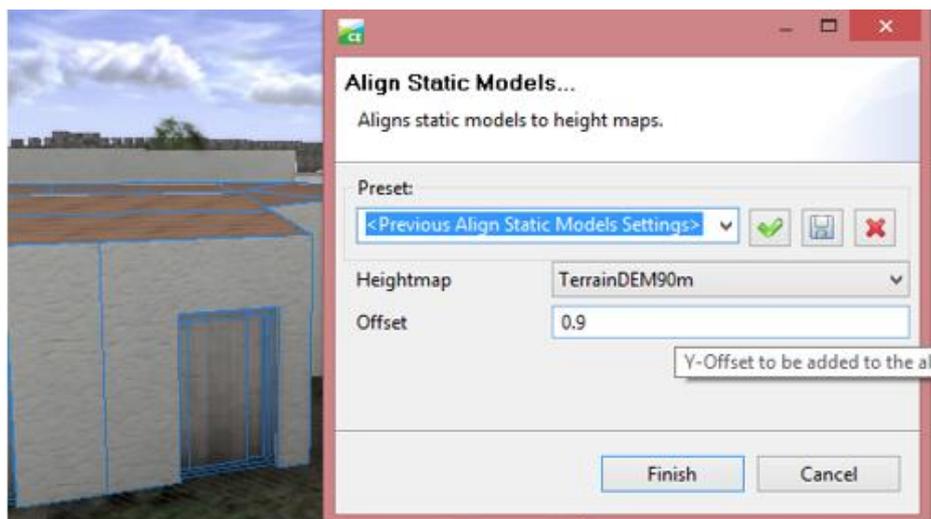


Figura 37 – Função alinhamento de modelos estáticos ao terreno.

4.5 Ambiente virtual interativo

O CityEngine permite a exportação do cenário tridimensional construído para ambientes interativos onde é possível navegar no espaço tridimensional simulando uma câmara e fontes de luz, entre outros recursos. Explica-se em seguida a exportação realizada para as plataformas CityEngine Web Scene e Unity 3D. Nesta última plataforma testou-se a integração de humanos virtuais que é um dos objetivos do projeto em que este trabalho se enquadra e que serviu de motivação para sua execução.

4.5.1 Exportação para ambiente virtual interativo

O CityEngine permite a exportação de modelos em diferentes formatos. Para Unity 3D há compatibilidade com os formatos OBJ e FBX e para o CityEngine Web Scene utiliza-se o formato 3WS, como ilustra na Figura 38. Nesta figura a seta contínua realça que o formato FBX é mais eficaz quanto à interoperabilidade com *unity game engine*, devido a este ser um formato 3D incorporado, o que significa que pode incluir malhas 3D, texturas e a técnica de mapeamento uvw map num único ficheiro. Por este motivo, torna-se viável aplicar *generate colliders* em toda a malha do projeto importado no ambiente Unity 3D.

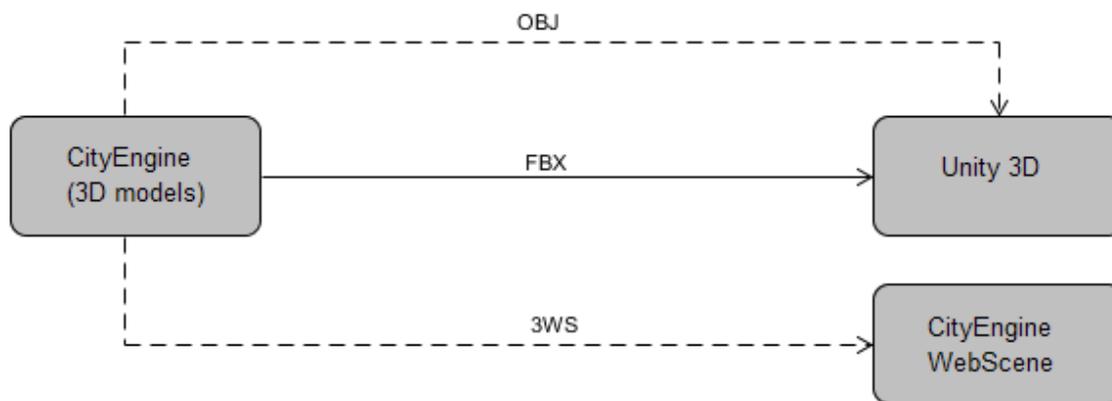


Figura 38 – Enquadramento da plataforma de exportação do CityEngine.

CityEngine Web Scene

O CityEngine Web Scene permite a publicação do conteúdo do ambiente urbano virtual na Internet. A importação para esta plataforma mantém a estrutura de camadas do CityEngine o que torna possível configurar os parâmetros de visualização de modo a seleccionar as camadas visíveis em cada instante. A Figura 39 mostra uma imagem do modelo no Web Scene. A barra inferior da imagem contém uma lista de pequenas imagens que têm ligações para pontos de vista pré-definidos e transportam o utilizador para essas posições da câmara. No anexo F encontram-se mais imagens do modelo no WebScene.



Figura 39 – Vista do CityEngine Web Scene.

Unity 3D

Para atender à animação de personagens virtuais no cenário *unity game engine*, houve especial cuidado na exportação do terreno e testaram-se duas abordagens distintas.

A primeira abordagem seguiu um processo análogo ao descrito por Saldaña e Johanson (2013) que exporta o terreno por imagem. Em primeiro lugar, o terreno é guardado num ficheiro .png, que é convertido para o formato *raw file*, usando o Photoshop. Depois recorre-se à opção "Terrain" do Unity 3D para importar este arquivo e associa-se um *collider* ao solo com a opção "*Terrain Collider*" do *Terrain Engine*⁵⁷. Esta solução levantou alguns problemas porque requer que *a posteriori* sejam ajustados ao terreno os modelos arquitetónicos, que são importados para o Unity 3D no formato FBX. O facto de se usarem duas formas de importação diferentes para o terreno e as estruturas arquitetónicas e por o terreno do caso de estudo de Mértola ser muito acidentado resultou num modelo final com muitas irregularidades. O artigo de Saldaña e Johanson (2013) mencionava já a necessidade de um ajuste manual neste processo, que, neste caso, se mostrou mais complexo devido à irregularidade do terreno.

A segunda solução de exportação consistiu em exportar o modelo 3D integralmente sem separar o terreno. Nesta abordagem gera-se um ficheiro no formato FBX, que contém todo o modelo. Para realizar esta operação, seguiram-se as instruções descritas nos documentos de apoio da ESRI⁵⁸, e, além disso, desativou-se a opção *Create shape group*⁵⁹. Esta solução melhorou o desempenho na exportação, pois foi muito importante para reduzir o número de *shapes* exportadas. Como resultado, houve uma redução significativa do tamanho do ficheiro e, simultaneamente, possibilitou a interação com a cena em tempo real no Unity 3D.

Na fase de importação para a plataforma *unity*, é seleccionada automaticamente uma variação de fatores de escala entre os dois sistemas.

A Figura 40 mostra o cenário do ambiente virtual de Mértola após exportação para o Unity 3D.

⁵⁷ Unity Documentation, "Terrain Settings", acessido a 06 de julho de 2016, em: <http://docs.unity3d.com/Manual/terrain-OtherSettings.html>

⁵⁸ ESRI, "How to: Export Textured Models from CityEngine into Unity", acessido a 13 de junho de 2016, em: <http://support.esri.com/technical-article/000012449>

⁵⁹ CityEngine, "General Export Reference", acessido a 06 de julho de 2016, em: http://cehelp.esri.com/help/index.jsp?topic=/com.procedural.cityengine.help/html/manual/export/export_formats_general.html

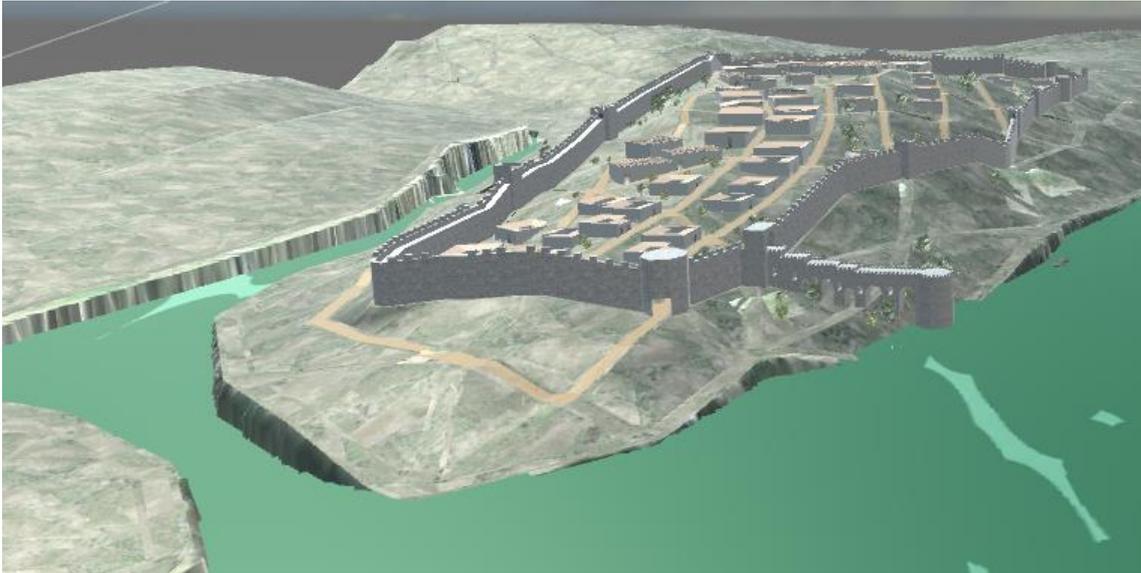


Figura 40 – Vista do Unity 3D do ambiente urbano virtual.

4.5.2 Personagem virtual interativo

Testou-se a inclusão de um personagem virtual no cenário importado no Unity 3D. Para permitir uma interação mais realista, aplicou-se a operação de *generate colliders*⁶⁰ na malha de toda a cena importada.

Para adicionar o personagem virtual interativo, recorreu-se aos ativos (*Assets Store*)⁶¹ da plataforma *unity*. Utilizaram-se os *prefabs* (que são instância de objetos) padrões criados e disponíveis num repositório de recursos gratuitos ou pagos da plataforma Unity⁶². Esta etapa permitiu a presença de uma personagem virtual com interação. Foram utilizadas as seguintes fases de importação de *Assets*:

- **Personagens virtuais:** Menu *Assets*, depois escolher *Import package*, em seguida *Characteres*;
- **Camera:** Menu *Assets*, depois escolher *Import package*, em seguida *Camera*.

Na cena virtual o personagem percorre o ambiente dentro muralha em diferentes direções conforme a escolha do utilizador. A sua movimentação é determinada pelo teclado. A Figura 41 mostra um humano virtual no cenário.

⁶⁰ Unity Documentation, “Importing Models”, acessido a 05 de julho de 2016, em: <http://docs.unity3d.com/Manual/FBXImporter-Model.html>

⁶¹ Unity, “Prototyping With Standard Assets”, acessido a 05 de julho de 2016, em: <https://unity3d.com/pt/learn/tutorials/modules/intermediate/live-training-archive/prototyping-standard-assets>

⁶² Unity, “Ajuda da Asset Store”, acessido a 05 de julho de 2016, em: <http://unity3d.com/pt/asset-store/help>



Figura 41 – Humano virtual no cenário.

4.6 Discussão

A investigação inicia-se com a identificação e definição das tarefas inerentes à reconstrução do modelo urbano virtual da vila histórica de Mértola. Um dos desafios foi a recolha de dados do modelo digital do terreno para a região de interesse do caso de estudo. Como já foi referido, é possível adquirir os dados à Direção Geral do Território, mas o sítio da *Open Topography*⁶³, disponibiliza dados gratuitamente. Atendendo ao relevo acidentado de Mértola, a granularidade da malha determinará o grau de aproximação do modelo gerado em relação ao terreno real. Contudo, é preciso ter em atenção que, se a malha for mais detalhada, geram-se mais polígonos, o que diminuirá o desempenho da interação em tempo real. A plataforma ArcGIS foi utilizada para interagir com outras ferramentas para: recolher dados vetoriais e de imagem; processar e integrar os dados de terreno e gerar os ficheiros de output para o CityEngine.

No desenvolvimento do trabalho foi necessário recorrer a diferentes ferramentas com propósitos distintos, como seja a modelação do terreno, modelação de edifícios, aplicação de texturas, integração dos modelos no mesmo ambiente e exportação para uma plataforma interativa com humanos virtuais. Houve por isso um desafio de compatibilidade entre as várias ferramentas, resolvido pela capacidade de exportação e

⁶³ Open Topography, “Find Lidar Topography Data”, acessido a 02 de junho de 2016, em: <http://opentopo.sdsc.edu/datasets>

importação de dados em formatos comuns, garantindo a interoperabilidade de dados 3D. Destacam-se os formatos FBX e OBJ, que oferecem o suporte para a definição do *pipeline* com maior eficiência, e ainda o formato 3WS utilizado para visualização no CityEngine *WebScene*. Através do formato FBX tornou-se possível exportar o modelo completo do ambiente urbano virtual, para que possa ser convertido em modelos de *GameObject* no *unity game engine*, atingindo assim o objetivo de construção do ambiente virtual interativo com a presença de personagens virtuais.

Outro desafio foi integrar a modelação de níveis de detalhe criados em diferentes ferramentas e a sua exportação para o Unity 3D, garantindo a variação de detalhe de acordo com a proximidade da câmara. Houve duas frentes distintas: o CityEngine e o 3DS Max. O CityEngine não possui um sistema implementado para computação direta, baseada na proximidade da câmara, mas permite criar diferentes níveis de detalhe para o mesmo objeto com inserção de *assets* com código de regras. Pelo contrário, o 3DS Max possui a funcionalidade do LOD, de acordo com a proximidade da câmara, e utiliza a ferramenta *level of detail utility* para criar grupos de objetos com diferentes níveis de detalhe, que podem ser exportados com eficiência para o Unity 3D. Acrescenta-se que na modelação tridimensional no 3DS Max houve um certo desafio para projetar medidas precisas, em contraste com a modelação no AutoCAD, mais utilizado em projetos de arquitetura.

Aproveitou-se a capacidade de modelação procedimental do CityEngine para agilizar a construção de modelos 3D. Contudo, em alguns casos isso não possível apenas com regras. Um exemplo, é o modelo das casas com telhado inclinado para o pátio interior. Tentou-se a modelação por código de regras, mas o resultado mais próximo que se obteve foram casas com tipologia açoteia, que de acordo com a avaliação pelos arqueólogos, não são da época Islâmica. Por isso, a solução mais automatizada que se obteve para esta tipologia consistiu em recorrer a técnica de modelação poligonal com combinação de código de regras (ver anexo G).

4.7 Conclusão

Neste capítulo apresentou-se um *pipeline* para a construção de ambientes urbanos virtuais que integra várias ferramentas, aproveitando a interoperabilidade de dados 3D entre elas e as características de cada uma delas como ferramentas de criação de conteúdo digital. A Figura 42 apresenta a visão global do *pipeline* resultante da

investigação desenvolvida para a reconstrução de espaços históricos englobando modelação manual e automatizada.

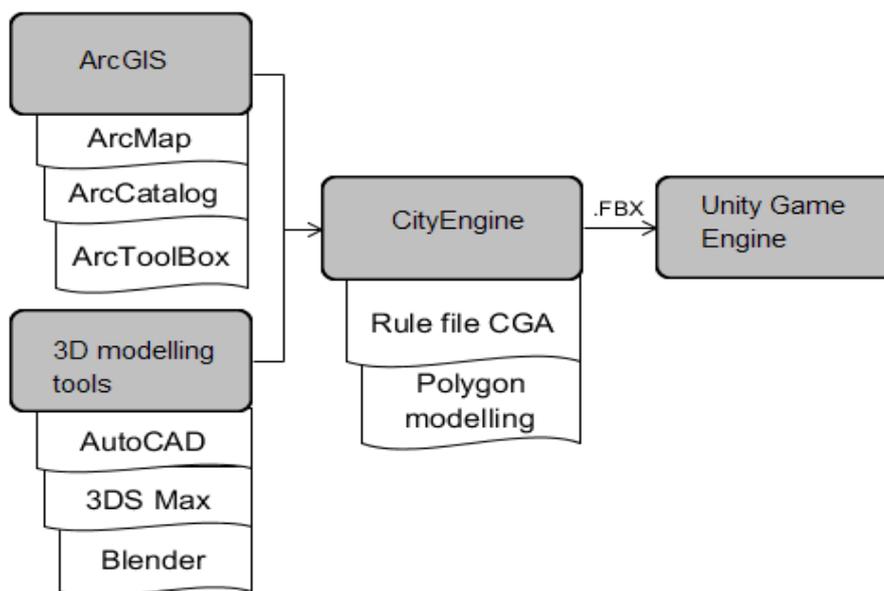


Figura 42 – Visão global do pipeline proposto.

Este *pipeline* integra:

- A plataforma ArcGIS, utilizando-se das ferramentas ArcMap, ArcCatalog e ArcToolBox na obtenção de dados geográficos e processamento de informação geográfica, para obter ficheiros de dados relativos ao modelo digital do terreno e imagem georreferenciada para textura.
- Ferramentas de modelação tridimensional com formatos de importação/exportação compatíveis entre si (OBJ ou FBX): AutoCAD, 3DS Max e Blender.
- O CityEngine como plataforma para integrar terreno e modelos tridimensionais sobre esse terreno, bem como, construir de raiz novos modelos, quer por modelação poligonal, quer por modelação procedimental.
- O Unity 3D como ambiente virtual interativo para onde transitou, no formato FBX, o cenário completo desenvolvido do CityEngine.

A validação do modelo desenvolvido contou com a opinião de arqueólogo que apresentaram sugestões de melhoramento incorporadas no modelo final. Além disso, o teste de inclusão de um humano virtual permitiu concluir que o modelo construído garante interação em tempo real. Mais ainda, o modelo foi utilizado por outros elementos de equipa para integrar grupos de humanos virtuais que se movimentam no espaço recorrendo a script com algoritmo de inteligência artificial

Capítulo 5

Conclusões e trabalho futuro

O resultado obtido com este trabalho é uma proposta de *pipeline* para a reconstrução digital de espaços históricos que foi concretizada num caso de estudo relativo à vila de Mértola no período de ocupação Islâmica. Recorreu-se a ferramentas gratuitas ou com licenças académicas de modo a minimizar os custos envolvidos e integraram-se modelos produzidos por vários alunos de licenciatura e de mestrado do DI-FCUL, em unidades curriculares dos seus cursos e como elementos de avaliação. Além desta integração foram ainda definidos outros modelos com recurso a modelação manual e procedimental. Ao longo de todo o processo houve a preocupação constante de encontrar um equilíbrio entre a qualidade dos modelos 3D e o número de polígonos, de modo a oferecer ao utilizador uma experiência credível mas que suportasse a sua interação em tempo real.

Validou-se o modelo final obtido: este suporta a inserção de personagens virtuais inteligentes e interação do utilizador em tempo real.

Sendo um trabalho no âmbito da herança cultural, foi necessário recorrer a informação fornecida por arqueólogos e efectuar pesquisas complementares. Deste modo, os modelos digitais produzidos seguiram vários tipos de fontes, como será expectável neste tipo de trabalhos.

Nas secções deste capítulo detalham-se os resultados obtidos, enumeram-se tópicos para trabalho futuro e apresenta-se uma perspectiva pessoal sobre a realização do trabalho.

5.1 Resultados obtidos

O modelo virtual de Mértola tem um terreno fiél ao terreno verdadeiro em que esta vila está implantada. O primeiro passo do *pipeline* é precisamente a obtenção deste

terreno, sobre o qual foram colocadas as construções virtuais, nos passos seguintes do *pipeline*.

A plataforma CityEngine foi usada para colocar sobre o terreno os modelos 3D produzidos previamente noutras ferramentas e também para gerar outros modelos usando modelação procedimental, tornando assim o procedimento de reconstrução semi-automatizado e permitindo a reutilização do processo noutros espaços históricos com características idênticas.

A Figura 43 mostra uma planta do cenário virtual e identifica as técnicas de desenvolvimento associadas a cada modelo ou conjunto de modelos. Enumeramos de seguida estes elementos, a forma como foram gerados e explicitamos a fonte arqueológica/histórica para os produzir:

a) Casas da Alcáçova: modelação manual em 3DS Max. Modelação baseada em informação textual, figuras e vestígios arqueológicos existentes na zona.

b) Casas genéricas 1: modelação manual em Blender e AutoCAD feita por alunos. Estes modelos foram importados para o 3DS Max para aplicação de textura. Algumas destas casas foram colocadas em locais específicos dentro das muralhas, enquanto outras foram usadas para geração procedimental de núcleos de casas dentro das muralhas e em zonas em que não existe informação exacta sobre a disposição das casas e ruas. Modelação baseada em informação textual e figuras.

c) Casas genéricas 2: modelação poligonal no CityEngine combinada com regras. Os modelos das casas foram desenvolvidos por modelação poligonal sendo posteriormente aplicado um código de regra para substituir as texturas, para inserir pátios interiores nas casas e telhados inclinados em direção ao pátio. A Figura G. 1 apresenta exemplo das casas geradas por esta técnica. Modelação baseada em informação textual e figuras.

d) Torre do Rio: modelação manual em AutoCAD feita por alunos. Este modelo foi integrado no cenário em CityEngine e foi-lhe aplicada a textura. Modelação baseada em informação textual, figuras e vestígios arqueológicos existentes na zona.

e) Muralha: modelação procedimental no CityEngine. O ponto de partida para obter o modelo da muralha foi a criação de shapes usando regras específicas para criação de ruas, recorrendo a um recurso do CityEngine. Depois de um ajuste manual destas shapes para melhorar o rigor da sua colocação sobre o terreno, foram aplicadas regras para gerar as paredes da muralha. Entre estas, nos nós das shapes, são colocadas

torres, previamente geradas por modelação poligonal. Modelação baseada em informação textual, figuras antigas, vestígios arqueológicos existentes na zona.

f) Vegetação: modelação procedimental usando a biblioteca de vegetação disponibilizada pela ESRI, por aplicação de um conjunto de regras para gerar textura na paisagem e modelos 3D de árvores. Usaram-se modelos de vegetação da zona do Baixo-Alentejo.

g) Ruas: modelação procedimental recorrendo a um recurso do CityEngine para geração de ruas. Adicionalmente foi aplicada uma regra para aplicação duma textura que simula terra batida. Modelação baseada em informação textual, figuras antigas, e traçado actual das ruas.

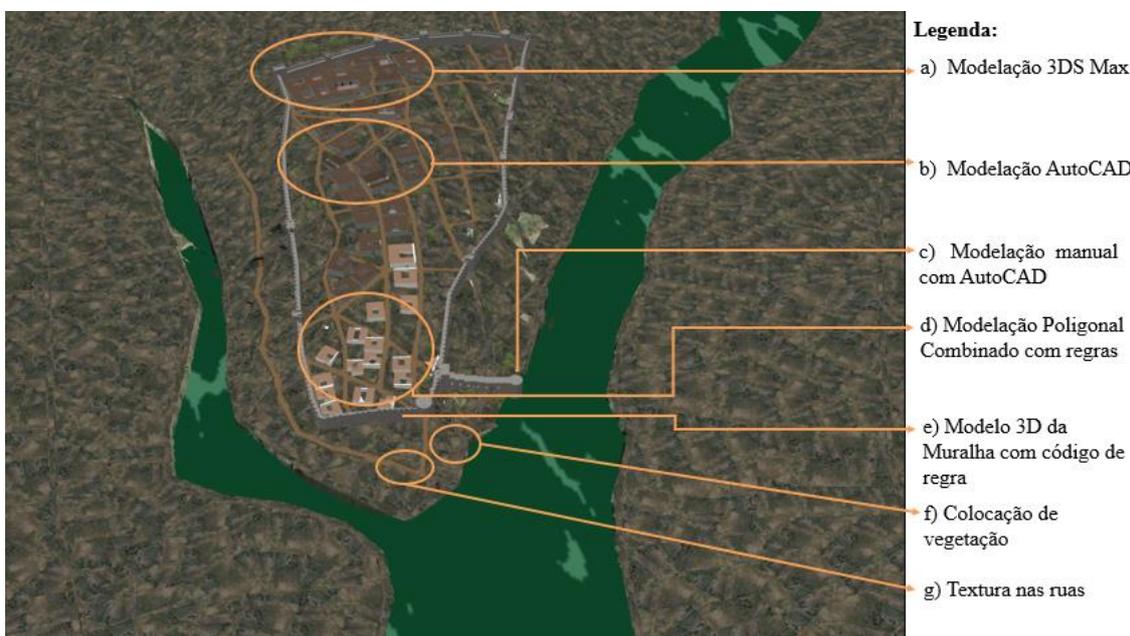


Figura 43 – Ambiente urbano virtual e técnica de construção.

Durante a fase de desenvolvimento, obteve-se uma outra versão do protótipo (Anexo B) que inclui casas do tipo açoteia que se caracterizam por possuir um telhado com parapeito. Para modelar estas casas foi utilizado um conjunto de regras. No entanto, na fase de validação pelos arqueólogos, decidiu-se eliminar do modelo estas casas porque esta tipologia não é característica nem da zona, nem da época que se quer recrear.

O modelo virtual de Mértola foi exportado para Unity 3D e foi validado com sucesso: suporta navegação interativa em tempo real mesmo com a inserção de humanos virtuais inteligentes.

5.2 Trabalho futuro

Alguns passos para trabalho futuro são identificados de seguida. Começando por tópicos relativos especificamente ao caso de estudo:

- Disponibilizar o modelo virtual de Mértola na Web, através de hospedagem e configuração de servidor web – *Web Viewer Scene*.
- Codificação de novas regras para gerar mais conteúdos arquitetónicos, por exemplo a ponte velha localizada sobre o rio Oeiras, ou outros que existam referidos em fontes arqueológicas/históricas.
- Aprimorar o código de regras com mais níveis de detalhes, que compreenda a importação de *assets*, por exemplo, telhas para as casas.

Num contexto mais abrangente, seriam interessantes os seguintes desenvolvimentos:

- Estudar outras formas de obtenção do modelo realista do terreno. Será importante estabelecer parcerias com entidades que fornecem modelos digitais do terreno. Por exemplo, se estiverem disponíveis as curvas de nível da topografia do terreno em formato AutoCAD digital, estas podem ser importadas pelo 3DS Max e neste software pode ser gerado o modelo 3D do terreno.
- A adaptação do *pipeline* para tornar possível a integração de elementos de três grandes épocas distintas: passado, presente e futuro. Isto levaria à integração no *pipeline* de outras ferramentas e métodos para obtenção de modelos existentes fisicamente no presente, por exemplo, fotogrametria e *laser scanning*.
- Estudar outras formas de obter a exportação do modelo virtual, possibilidade de desenvolver automatização para proceder com a exportação do modelo 3D do CityEngine no formato FBX.

5.3 Conclusão

A presente investigação teve como principal característica a superação de desafios, que foram apresentados desde o início, devido a uma variedade de ferramentas envolvidas com características específicas (CityEngine, ArcGIS, 3DS Max, Unity 3D AutoCAD, Blender, entre outras), e que exigiu a superação através da busca pelo conhecimento destas ferramentas quanto ao desenvolvimento de modelos 3D, para integração em um *pipeline* sem perda de dados 3D, o que possibilitou a abordagem de

construção de ambiente urbano virtual. A temática herança cultural digital exigiu a procura de informações históricas e arqueológicas através de literatura. O que tornou motivador foi representar este resgate histórico, um caso de estudo da vila histórica de Mértola num período entre os séculos VII e XIII, numa plataforma virtual. Apesar das dificuldades de simular a realidade com alta precisão em ambiente virtual, devido à complexidade de aquisição do terreno da área de interesse, foi gratificante realizar a simulação mais precisa possível da realidade de um momento histórico do período Islâmico no território de Portugal.

A dinâmica desenvolvida em equipa enriqueceu sobremaneira o trabalho, através de partilha de ideias, sugestões de melhoramento e críticas relevantes em todas as fases de desenvolvimento da investigação. Os testes de validação possibilitaram interação entre as partes envolvidas, o que considero extremamente importante para identificar problemas e também soluções para resolver um determinado contratempo, por exemplo, deparámo-nos com a problemática de compatibilidade de dados 3D na exportação do cenário virtual. Acrescenta-se como facto positivo a elaboração de dois artigos científicos no decorrer do trabalho, que contribuíram para reforçar e esclarecer as etapas de desenvolvimento do ambiente urbano virtual. Regista-se o acompanhamento contínuo do trabalho realizado pelas orientadoras, com reuniões semanais que contribuíram para mapear as dificuldades enfrentadas e definição de estratégias para superação de desafios quanto a utilização das ferramentas para alcançar o melhor resultado possível.

Anexo A: Documentação técnica de geração do terreno e imagem

Este anexo complementa a secção 4.1 geração do terreno e possui como objetivo informações pormenores para proceder com a recolha de dados de modelo digital do terreno e imagem de satélite através do Google Earth.

A.1: Aquisição de modelo digital do terreno

Consiste em *raster dataset* para processamento do modelo digital do terreno de Mértola. O sítio da *Open Topography* disponibiliza modelo digital do terreno no formato GeoTiff. Os modelos digitais do terreno correspondem em dados real do terreno, que possuem células que representam os dados de altura (Zhou, et al. 2007).

Os dados obtidos pela *Open Topography* correspondem a cerca de 90 metros ou (295 pés) do projeto internacional *Shuttle Radar Topography Mission (SRTM)*, liderado pela *National Aeronautics and Space Administration (NASA)*.



Figura A. 1 – Aquisição de modelo digital do terreno.

A.2: Pormenor para aquisição e processamento de dados

Título	<i>Pipeline ArcGIS – ArcMap</i>
Visão geral	Plataforma de processamento de informação geográfica. O documento possui como objetivo oferecer suporte para aquisição de dados <i>raster dataset</i> e o processamento em ArcGIS-ArcMap para gerar ficheiros de input para o modelo digital do terreno e imagem georreferenciada para o ambiente CityEngine.
Instalação versão avaliação	<ol style="list-style-type: none"> <li data-bbox="472 506 1378 573">1. ArcGIS for Desktop (ArcMap): http://www.esri.com/software/arcgis/arcgis-for-desktop <li data-bbox="472 573 1378 707">2. ArcGIS Data Interoperability: http://www.esri.com/software/arcgis/extensions/datainteroperability
Requisitos para aquisição de fonte de dados	Fonte de dados para manipulação: <ol style="list-style-type: none"> <li data-bbox="472 752 1378 864">1. Modelo digital do terreno: aceder <i>Open Topography</i> para aquisição de dados de área de interesse: http://opentopo.sdsc.edu/datasets. <li data-bbox="472 864 1378 976">2. Aceder Catálogo de Geo Webservices, opção lista de serviços WFS – <i>Web Feature Service</i>, que disponibiliza dados geográficos em formato vetorial: http://mapas.dgterritorio.pt/ <li data-bbox="472 976 1378 1061">3. Imagem de textura: aceder o Google Earth Pro para aquisição de imagem, após criação de <i>shapefile</i> ponto de controle no ArcMap.
Referências	<ol style="list-style-type: none"> <li data-bbox="472 1061 1378 1173">1. Guia de apoio a utilização de serviços WFS: http://mapas.dgterritorio.pt/Openviewer/cos2007/Guia_de_apoio_utilizacao_WFS_ArcGIS.pdf <li data-bbox="472 1173 1378 1285">2. Cosme, A. (2012), “Projeto em Sistemas de Informação Geográfica”. Lisboa: Editora Lidel – Edições Técnica Ltda, 2012. <li data-bbox="472 1285 1378 1397">3. Creating a mosaic dataset: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/creating-a-mosaic-dataset.htm

Parte 1: Criar imagem da área de interesse georreferenciada:

1. No ArcCatalog, realizar a conexão com WFS camada (*layer*), disponível pela DG Território (ver material de guia de apoio da ESRI). Carregar os dados de mapa do território de Portugal no ambiente ArcMap. Após concluída as etapas de configuração do serviço presente no guia de apoio de utilização de serviços WFS, ele torna-se pronto para ser utilizado.
 - a. Arrastar *shapefile* polígonos Concelhos-2015 para o ambiente ArcMap para criar uma camada (*layer*) Concelhos-2015.
2. No ArcCatalog, criar uma *shapefile* do tipo “*point*” ponto de controlo. O processo de criação de *shapefile* é composto por três passos: 1) A escolha do nome da *shapefile*, a escolha do tipo de geometria (linha, ponto ou polígono) e c) a definição de sistema de projeção. A extensão de um ficheiro *shapefile* é

- .shp**, no entanto o objeto *shapefile* é composto por vários ficheiros (**.dbf**, **.prj**, **.sbn**, **.sbx**, **.shp**, **shp.xml** e **.shx**).
- a. No ArcCatalog, aceder *Create new shape file*, escolher opção *point* em *Feature type*. Definir o Sistema de coordenada (para o caso de estudo, define-se: WGS 1984 UTM Zone 29N).
 - b. Arrastar *shapefile* “Ponto de Controlo” criado para o ambiente ArcMap para criar uma camada (*layer*).
3. No ArcMap, realiza-se os seguintes procedimentos:
- a. Aceder Editor -> *start editing*. Após finalizar os quatro (4) pontos de controlo para área de interesse, clicar em *stop editing*.
 - b. Selecionar camada (*layer*) e clicar botão direito *add field*. Os campos, a título de exemplo, serão *coord_x* e *coord_y*, tipo de dados *double*.
 - c. Depois dos campos criados, aceder tabela e clicar botão direito na coluna campo *coord_x* e escolher a opção *calculate geometry* para coluna *coord_x* e *coord_y* criadas. Os valores serão *units* em metros para os pontos de controlo. Nesta etapa, o ArcMap realiza a conversão para metros.
 - d. Selecionar camada (*layer*) *point of control* criado e aceder caixa de ferramenta ArcToolBox e escolher *Conversion Tools* -> To KML -> Layer to KML. Em seguida preencher os parâmetros de output para *Google Earth*.
 - e. No *Google Earth Pro*, importa o ficheiro KMZ para guardar imagem da área de interesse (ver Figura A. 2). Após download da imagem arrastá-la para o ambiente ArcMap.
4. Etapa de georreferenciamento da imagem. Após a *shapefile* ponto de controlo criado e imagem da área de interesse adquirida via *Google Earth Pro*, proceder com as seguintes etapas para o georreferenciamento:
- a. Antes de ativar a ferramenta *Georeferencing*, clicar em *view link table* e desabilitar *Auto Adjust*.
 - b. Alternar entre as camadas (*layers*) imagem com apontamentos em destaque de pontos de controlo e camada (*layer*) *shapefile* de ponto de controlo para realizar o mapeamento das coordenadas na imagem através da ferramenta *Georeferencing*.
 - c. Na ferramenta *Georeferencing*, após mapeamento dos quatro pontos de controlo, clicar em *Rectify* para validar imagem com dados de georreferenciamento. Em seguida, escolhe-se o formato JPG e guardar a imagem georreferenciada. Após conclusão de validação da imagem, arrastá-la para o ambiente ArcMap.

Parte 2: Criar Área de Interesse e Processar modelo digital do terreno

1. Criar área de interesse com a ferramenta de desenhar polígono sobre a imagem georreferenciada. Depois, clicar com botão direito na ferramenta *Drawing*, e escolher a opção *Convert Graphics to features* (consiste em converter gráfico em *feature* para serem armazenados em *geodatabase*), para em seguida, na opção *output shapefile or feature class*, clicar para criar nova diretoria e *new file Geodatabase*.

- a. Designar como Mertola_3D – e efetivar com duplo click.
 - b. Logo em seguida, na opção *name*: denominar de **AIOMertolaV** – área de interesse e ativar opção *Automatically delete graphics after conversion*.
2. Na opção *search*, digitar *mosaic to new raster*, ferramenta de processamento para utilizar como objetivo de adicionar *raster* da imagem georreferenciada para área de interesse criado na etapa anterior.
 - a. Na interface *Mosaic to new raster*, arrastar *raster* da imagem georreferenciada de *table of content* para *input raster* (ver Figura A. 3, número 1).
 - b. Em *output location*, adicionar *dataset* à diretoria (por exemplo, c:\Mertola).
 - c. Em *raster dataset name*, designar o nome com extensão: (por exemplo, AOIMertola.tif).
 - d. Definir *spacial reference*, para o caso de estudo utiliza-se WGS_1984_UTM_Zone_29N.
 - e. 8 bit de profundidade.
 - f. *Number of bands* igual a 3. Após esta etapa, clicar em *next* para preencher parâmetros para **Environment Settings** (Figura A. 3, número 2).
 - g. *Environments settings*, opção *output coordinates*, definir a coordenada.
 - h. Em *processing extent*, escolher a camada (*layer*) **AIOMertolaV**.
 - i. Clicar em *ok*, processo finalizado para imagem georreferenciada.
3. Adicionar o ficheiro do modelo digital do terreno (output_srtm.tiff) no ambiente ArcMap para realizar o processamento.
 - a. Aceder **ArcToolBox** e selecionar caixa de ferramenta *Spatial Analyst Tools -> Map -> Raster Calculator*.
 - b. Em *output raster*, direcionar diretoria do ficheiro e nomear como “DEM90m.img”, em seguida, **Environment settings** definir os seguintes parâmetros:
 - c. *Output coordinates*: selecionar camada (*layer*) do modelo digital do terreno (output_srtm.tiff).
 - d. *Processing extent* (ver Figura A. 4 em destaque): *Same as layer*: escolher a camada (*layer*) **AIOMertolaV**, que consiste na área de interesse criada.
4. O resultado do processamento do modelo digital do terreno será o ficheiro DEM90m.img definido anteriormente e direcionado na diretoria.

Parte 3: Exportar imagem georreferenciada:

1. Após as camadas (*layers*) serem criadas, torna-se praticável exportar os ficheiros relacionado a imagem georreferenciada: clicar com botão direito da camada (*layer*) imagem georreferenciada e escolher opção *Data-> Export Data*. Na caixa de diálogo, definir tamanho para *Raster Size (Columns e rows)* e formato *JPG*. (por exemplo, 4000x4000).

A.2: Aquisição de imagem de alta resolução

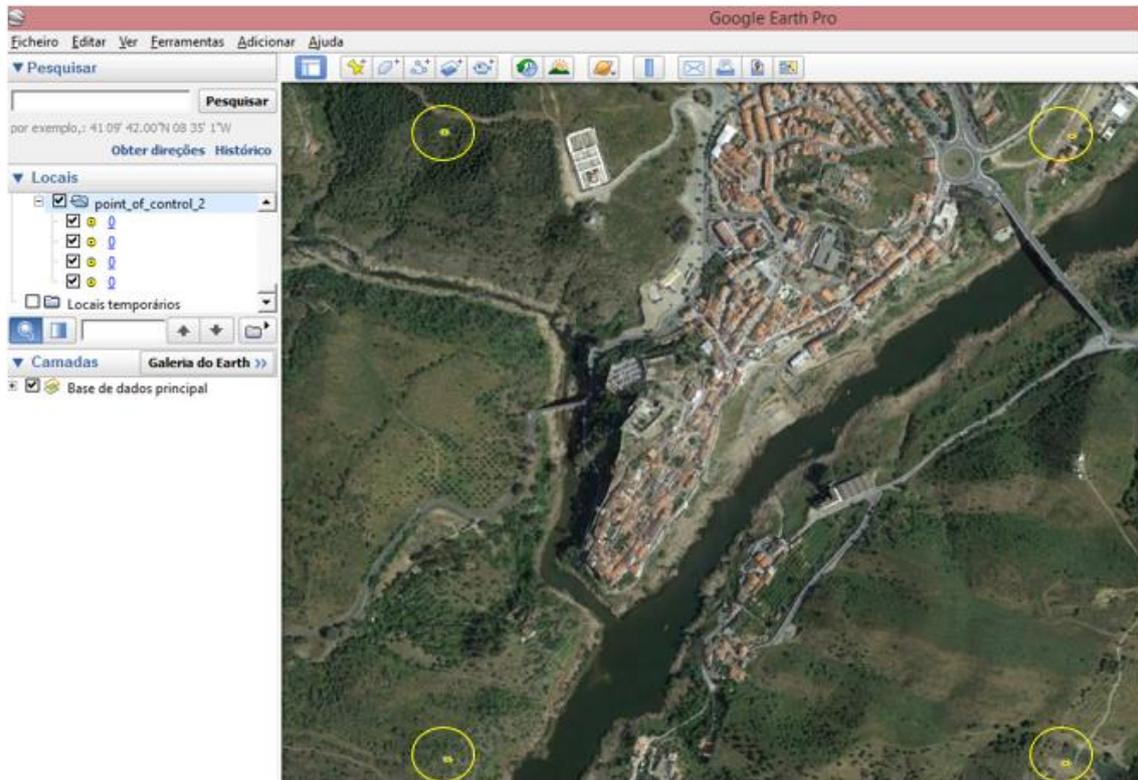


Figura A. 2 – Aquisição de imagem de alta resolução, assinala as coordenadas que limitam a área de interesse.

A.3: Parte da etapa da criação da área de interesse

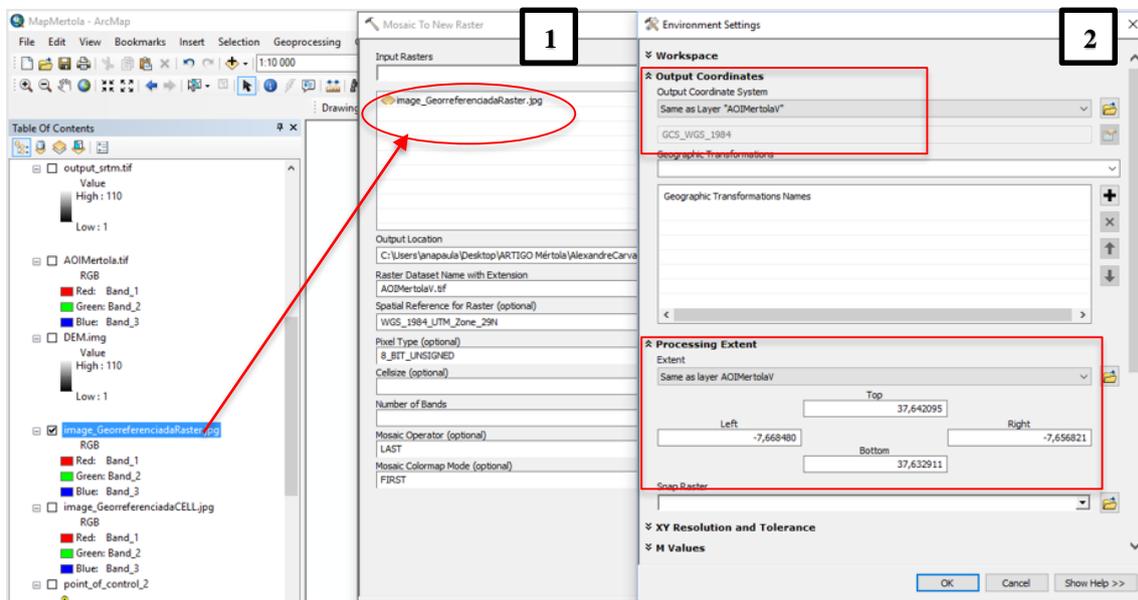


Figura A. 3 – Criação da camada área de interesse que servirá de apoio para recortar a imagem georeferenciada e modelo digital do terreno.

A.4: Parte da etapa de processamento do modelo digital do terreno

A opção *processing extent*, correspondem relacionar a camada de área de interesse criada com a definição de coordenada para os campos: *Top*, *Left*, *Bottom* e *Right*.

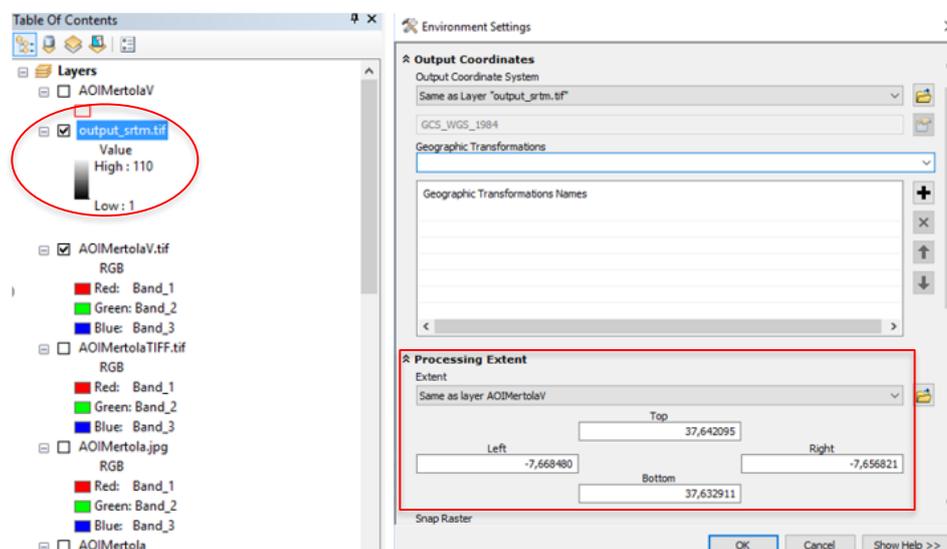


Figura A. 4 – Etapa do processamento do modelo digital do terreno, lado direito o ficheiro da fonte de dados Open Topography selecionado e em destaque lado direito a camada área de interesse definida para recortar a imagem e modelo digital do terreno.

A.5: Ficheiro de imagem georreferenciada

O ficheiro .jpw (AOIMertolaVirtual.jpw) é um ficheiro do tipo *World File*. Este ficheiro georreferencia a imagem, e encontra-se na mesma diretoria da imagem (Figura 13, lado esquerdo, topo). Segundo (Cosme, 2012, p. 194): “*World file* é um arquivo ASCII comum, armazenado na mesma pasta (diretoria) da imagem, e que possui o mesmo nome e extensão (de três letras), seguindo a regra: 1ª letra da extensão, última letra da extensão e W de *World*”. A estrutura deste ficheiro apresenta-se seis linhas, conforme a descrição pelo autor na Tabela 19.

Dados do ficheiro	Descrição
0,4014164519	Uma primeira linha com a dimensão de um pixel em unidades na direção x (resolução).
0,0000000000	A segunda linha com o valor de translação.
0,0000000000	A terceira com o valor de rotação.
-0,4014164519	A quarta com o valor de dimensão do pixel em y, em unidade de direção y – tem o sinal negativo, pois a medida faz-se de cima para baixo no sentido contrário ao crescente do eixo dos yy
-7,6742685014	A quinta linha tem as coordenadas de x do ponto de inserção (corresponde ao meio do pixel do canto superior esquerdo).
37,6445790706	A sexta linha tem as coordenadas de y do ponto de inserção (corresponde ao meio do pixel do canto superior esquerdo).

Tabela 19 – Descrição do ficheiro world file da imagem georreferenciada.

A.6: Área de interesse – Vila de Mértola

Usou também o visualizador do Sistema Nacional de Informação Geográfica – SNIG (<http://snig.dgterritorio.pt/geoportalMapView/>) somente para ter uma noção da área de interesse, onde desenhou um polígono para identificação da localidade do caso de estudo.



Figura A. 5 – Localização da Área de interesse para o caso de estudo Mértola Virtual.

Anexo B: Ambiente urbano virtual



Figura B. 1 – Cenário virtual com casas tipo açoteia gerada por modelação procedimental, em destaque no lado direito da figura, em cima e em baixo.



Figura B. 2 – Casas do tipo açoteia em destaque na cor laranja no cenário virtual.

Anexo C: Fases intermediária de reconstrução do ambiente urbano virtual

Este anexo complementa a secção resultados obtidos e possui como objetivo apresentar as fases de evelução do ambiente urbano virtual.



Figura C. 1 – Fase com importação da muralha em objeto 3D.



Figura C. 2 – Muralha com ficheiro de regra.

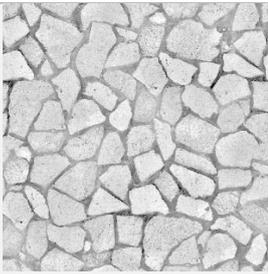


Figura C. 3 – Cena virtual sem a melhoria de textura da muralha.



Figura C. 4 – Vista final do modelo virtual com melhoria de textura da muralha e uma camada para simular os rios Oeiras e Guardiana.

Anexo D: Texturas aplicadas na modelação com regras

Código de regra (extrato)	Objetivo	Textura
<pre>// Courtyard of house Courtyards --> set(trim.vertical, true) alignScopeToAxes(y) split(y) {~courtyardHeight:split(x) {~courtyardWidth: Courtyard}*}* Courtyard --> 1, 0) texture("house-2/maps/pedral.jpg") projectUV(0)</pre>	Aplica textura no pátio interno da casa.	
<pre>Walls --> set(trim.vertical, true) alignScopeToAxes(y) split(y) {~wallHeight:split(x) {~wallWidth: Wall}*}* @Location(2341,97) Wall --> setupProjection(0, scope.xy, 3.5, 1.5, 1, 0) texture("house-2/maps/parede_casa_branco_2.jpg") projectUV(0)</pre>	Aplica textura na parede da casa.	
<pre>Roofs --> set(trim.vertical, false) alignScopeToAxes(y) split(y) {~roofHeight:split(x) {~roofWidth: Roof}*}* Roof --> setupProjection(0, scope.xy, 1, 0.2, 0.2) texture("house-2/maps/Roof.Tiles.jpg") projectUV(0)</pre>	Aplica textura no telhado da casa.	

<pre> Doors --> set(trim.vertical, true) alignScopeToAxes(y) split(y){~doorHeight:split(x){~doorWid th: Door}*}* // Assets insertion OBJ format Door --> i("door/door-2_from_CE_0.obj") s(1.9,'1,0.2) t(-0.3,0,'-0.2) mirror(true,false,false) </pre>	<p>Insera textura na geometria da porta da casa.</p>	
---	--	---

Tabela D. 1 - Textura com modelação poligonal e conjunto de regras.

Código de regra (extrato)	Objetivo	Textura
<pre>/*File: ruleStreets.cga*/ //Street Texture para dirtmap const street_texture "landscape/street/street_06.jpg"</pre>	<p>=</p> <p>Aplicar textura sobre a <i>shape street</i> – consiste em definir textura para ruas para o caso de estudo.</p>	
<pre>/* File: ruleLandscape.cga*/ // Texture landscape randFieldTexture fileRandom("landscape/texture_03.jpg")</pre>	<p>=</p> <p>Aplicar textura sobre <i>shape</i> inerente a paisagem com vegetação.</p>	
<pre>/* File: ruleWallCity.cga*/ // Texture vertical WallVerTexturing --> WallVerTexturingSelectTexture(file Random("wall/wallTower_02.jpg")) // Texture Horizontal WallHorTexturing --> WallHorTexturingSelectTexture(file Random("wall/wallTower_02.jpg"))</pre>	<p>Aplicar textura na muralha e torre do rio.</p>	
<pre>// Texture for tower const gateTexturing = "3D Models/torreao/gateWall.jpg"</pre>	<p>Aplica textura para similar porta na torre.</p>	
<pre>// Map Layer com ficheiro de textura do // rio MertolaAOI/AOIMertolaVirtualRiver.png</pre>	<p>Aplica textura na camada (<i>layer</i>) rio criado no CityEngine.</p>	

Tabela D. 2 - Textura por modelação procedimental.

Anexo E: Modelação 3D com LODs



Figura E. 1 - Vista geral das 15 casas da Alcáçova.

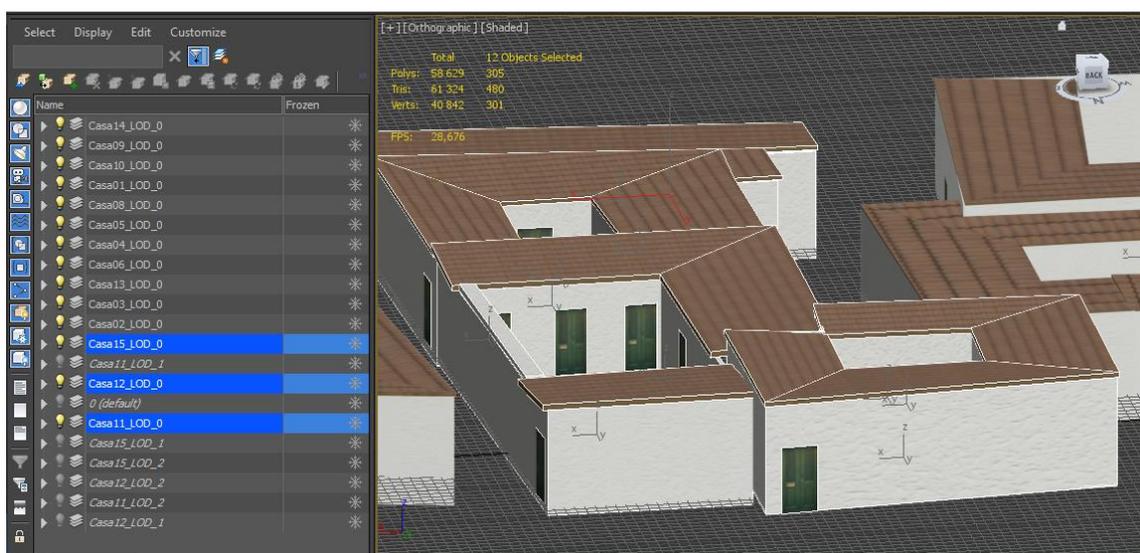


Figura E. 2 – Casas XI, XII e XV, LOD_0 com 305 polígonos.

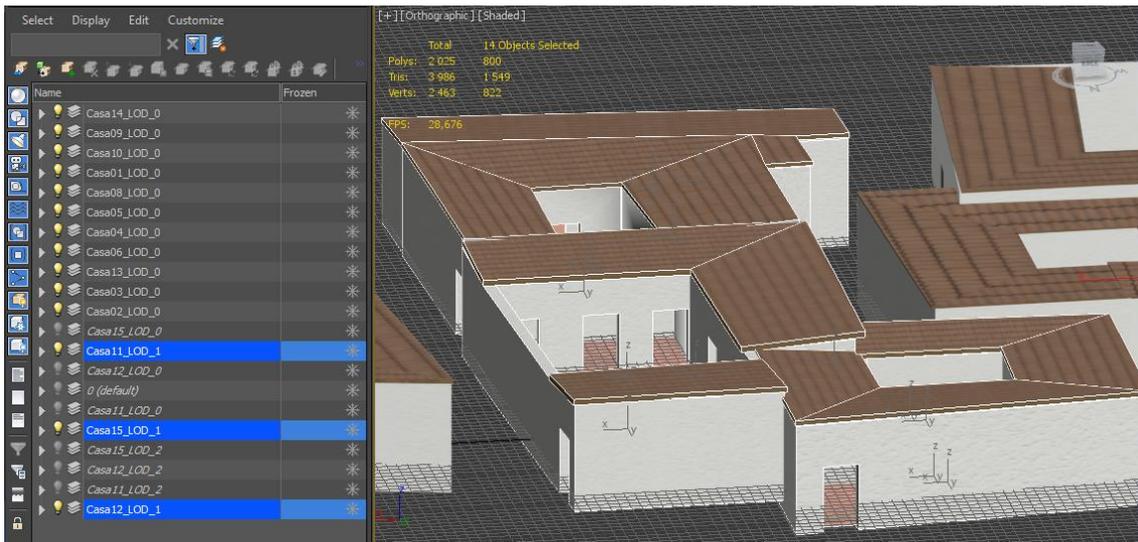


Figura E. 3 – Casas XI, XII e XV, LOD_1 com 800 polígonos.

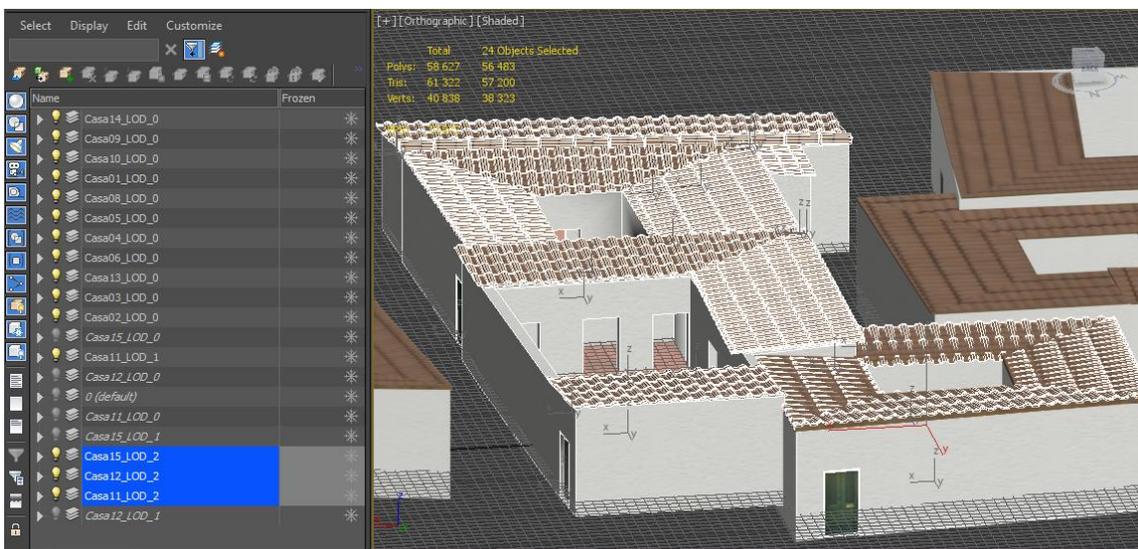


Figura E. 4 – Casas XI, XII e XV, LOD_2 com 56.483 polígonos.

Anexo F: Visualização do ambiente virtual urbano no CityEngine e Unity



Figura F. 1 – Vista 2 do CityEngine web scene.

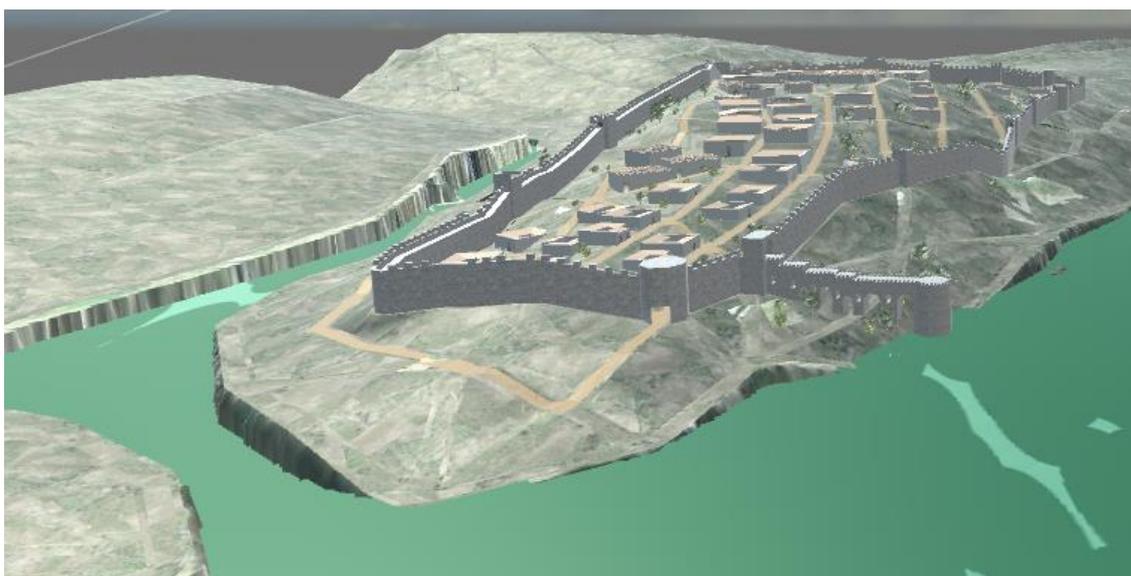


Figura F. 2 – Vista do Unity 3D do ambiente urbano virtual.

Anexo G: Casas geradas com combinação de regras e modelação poligonal

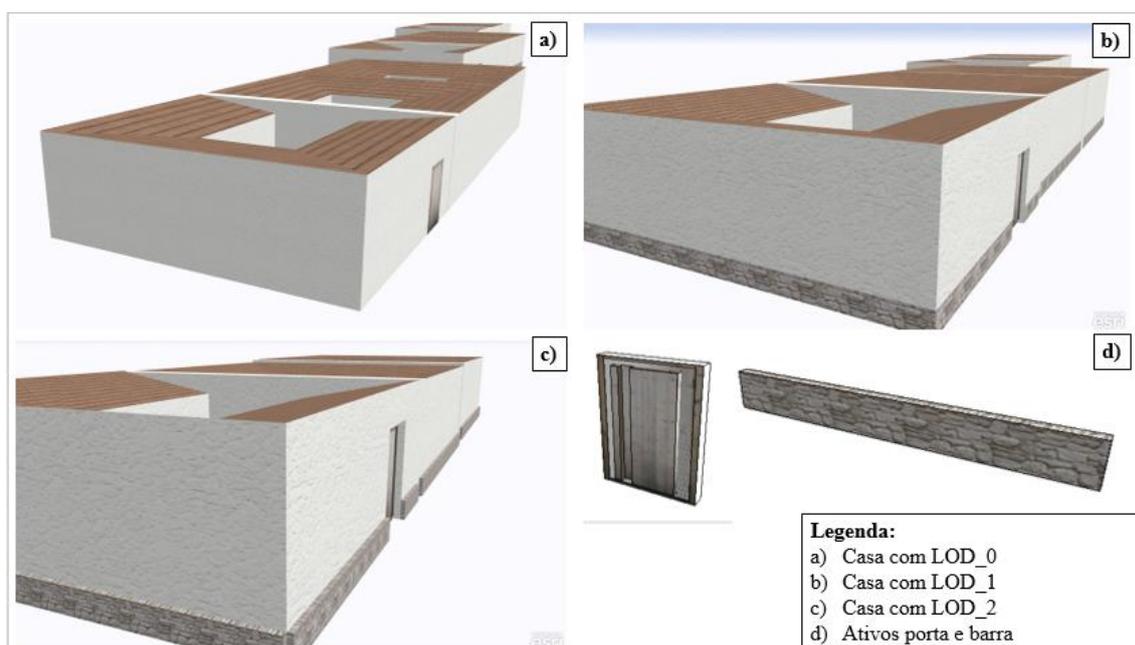


Figura G. 1 - Casas geradas com combinação de modelagem poligonal e regras.

Anexo H: Princípios da carta de Londres e

Sevilha

A carta de Londres, publicada em fevereiro de 2009, possui como objetivo estabelecer uma série de princípios inerente a visualização computadorizada no campo da investigação e divulgação do patrimônio cultural por tecnologia da informação. Os princípios abordados são: implementação; objetivos e métodos; fontes de investigação; documentação; sustentabilidade e acesso. Possui como âmbito, assegurar rigor científico e intelectual de projetos de reconstrução de herança cultural digital.

A carta Sevilha foi estabelecida em julho de 2010 com objetivo de assegurar a aplicabilidade dos princípios (interdisciplinaridade, propósito, complementaridade, autenticidade, rigor histórico, eficiência, transparência científica e treinamento e avaliação) de recomendações da carta de Londres no campo da arqueologia virtual. O âmbito da carta estabelece a implementação destes oito princípios a serem seguidos com rigor quanto ao patrimônio arqueológico em relação ao patrimônio cultural.

Acrescenta-se ao contexto de historicidade das cartas as atualizações para adequação diante dos avanços tecnológicos na temática herança cultural digital. Com chegada da tecnologia e avanço contínuo, torna-se possível criar herança cultural digital baseada em computadores, é o que se constata no trabalho de (Denard, 2012). O trabalho consiste em uma nova introdução a carta de Londres, com objetivo específico de contemplar arqueologia virtual. Pois herança cultural digital baseada em computador, resgata-se o passado e estende-se o presente representado por computador.

A Tabela H. 1 apresenta o cronograma histórico e comparação das recomendações específicas e as cartas, registrando-se o âmbito de cada tratado e convenções:

Ano de publicação	Nome do Documento	Âmbito
1972	Tratado sobre a convenção para proteção de património mundial, cultural e natural.	Definição e proteção internacional do património cultural e natural.
2003	Carta de preservação de herança digital.	Herança digital consiste em recursos de expressão e conhecimento humano.
2009	Carta de Londres	Define princípios para o uso de método de visualização computadorizada. Assegurar rigor técnico e intelectual.
2010	Carta de Sevilha	Assegurar a aplicabilidade dos princípios de recomendações da carta de Londres no campo da arqueologia virtual
2012	Carta de Londres	Atualização da carta – acrescenta uma nova introdução ao capítulo 6, para contemplar a carta de Sevilha, que cobre arqueologia virtual.

Tabela H. 1 – Cronologia de histórico de convenções.

Os princípios definidos na carta de Londres e Sevilha, correspondem a eficiência e eficácia quanto a disponibilidade de herança digital cultural disponível aos utilizadores.

Tendo em vista o avanço tecnológico, e a preocupação com património cultural digital, a comunidade de cientistas e poder público sugerem as regulamentações e recomendações, o que resulta o surgimento das cartas de Londres e Sevilha.

Bibliografia

Abbagnano, N. (2012), “Dicionário de Filosofia”. São Paulo: Editora WMF Martins Fontes, 2012.

Armas, D. (1509), “Livro das Fortalezas Situadas no Extremo de Portugal e Castela por Duarte de Armas, Escudeiro da Casa do Rei D. Manuel I”, acessado a 22 de novembro de 2015, em: <http://digitarq.dgarq.gov.pt/details?id=3909707>

Alisson, C. & Miller, A. (2012), “Open virtual worlds for open learning”. *The Higher Education Academy*, 2012.

ArcGIS Help 10.2, “What is raster data?”, acessado a 31 de março de 2016, em: http://resources.arcgis.com/en/help/main/10.2/index.html#/What_is_raster_data/009t0000002000000/

Barazzetti, L., *et al.* (2015), “HBIM and augmented Information: towards a wider user community of image and range-based reconstructions”. *ISPRS Archives - 25th International CIPA Symposium*, 2015, Vol. (XL-5/W7), pp. 35 – 42.

Bass, L. (2013). “Software Architecture in practice” (3rd ed.). SEI series in software engineering.

Besuievsky, G. & Patow, G.(2013), “An architectural approach to efficient 3D urban modeling”. *VAR. Volumen 4*, Número 9, pp. 160 – 166.

Carmo, M.B. & Cláudio, A.P. (2013), “3D Virtual Exhibitions”. *Journal of Library & Information Technology*, DESIDOC 2013, Vol. 33, nº. 3, May 2013, pp. 222 – 235.

CityEngine TV, “CityEngine Essential Skills”, acessado a 09 de julho de 2016, em: https://www.youtube.com/playlist?list=PLWGP11THb9EpvmuokMrj_yJ6L4qCuKVli.

Cláudio, A., *et al.* (2013), “The Teaching Astronomical Observatory of the University of Lisbon (19th century): A Virtual Experience”. *DigitalHeritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 367 – 370.

Cosme, A. (2012), “Projeto em Sistemas de Informação Geográfica”. Lisboa: Editora Lidel – Edições Técnica Ltda, 2012.

Deterding, S., *et al.* (2011), “From game design elements to gamefulness: defining gamification”. *MindTrek '11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, ACM, pp. 9 – 15.

Dylla, K., *et al.* (2010), “Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques”. *CAA 2009, Making History Interactive. 37th Proceedings of the CAA Conference March 22-26, 2009, Williamsburg, Virginia* (Archaeopress: Oxford, 2010), pp. 62-66.

Earthy, J. (2012), “ISO standards for user centered design and the specification of usability”, acessado a 02 de outubro de 2015, em:

- <http://www.nigelbevan.com/papers/ISO%20standards%20for%20user%20centered%20design%20and%20the%20specification%20of%20usability%20-%20submitted.pdf>.
- Fläckman, J., Roos, J. (2006), “Animationsanpassad mjukvarupipeline för Motion Capture leverantörer”. *LTU Skellefteå*, 2006.
- Fouts, J. (2010), “Al-Andalus 2.0”. *Saudi Aramco World*, 2010, pp. 10-15.
- Gabellone, F., *et al.* (2013), “From museum to original site: A 3d environment for virtual visits to finds recontextualized in their original setting”. *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 215 – 222.
- Gleue, T., & Dähne, P. (2002), “Design and Implementation of a Mobile Device for Outdoor Augmented Reality in the ARCHEOGUIDE Project”. *VAST '01 conference of Virtual reality, archeology, and cultural heritage, 2001*, pp. 161 – 168.
- Gomes, R. V., (2002), “Silves (Xelb), uma cidade do Gharb Al-Andalus: Território e Cultura”, Instituto Português de Arqueologia, Lisboa, 2002, Trabalho de Arqueologia, 23.
- Haas, F., *et al.* (2013), “Dresden City Models: On the interrelation of virtual reconstructions and the image of a city”. *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 99 – 102.
- Haegler, S., *et al.* (2009), “Procedural Modeling for Digital Cultural Heritage”. *EURASIP Journal on Image and Video Processing*, 2009, Vol. 2009, 11 pages.
- Heimgärtner, R. (2014), “ISO 9241-210 and Culture? – The Impact of Culture on the Standard Usability Engineering Process”, *A. Marcus (Ed.): Duxu 2014, Part. IV, LNCS 8520*, pp. 39 – 48.
- Jaeger, W. W. (2013), “Paidéia: a formação do homem grego”. 6°. Ed. – São Paulo: Editora WMF Martins Fontes, 2013.
- Kelley, D., Malin, C., Nielson, M. (1988), “Terrain simulation using a model of stream erosion”. In *IGGRAPH'88: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA, 1988)*, ACM, pp. 263–268.
- Kennedy, S., *et al.* (2013), “Exploring Canons & Cathedrals with Open Virtual Worlds The Recreation of St Andrews Cathedral, St Andrews Day, 1318”. *DigitalHeritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 273 – 280.
- Lercari, N., Forte, M., Onsurez, L. (2013), “Multimodal Reconstruction of Landscape in Serious Games for Heritage An insight on the creation of Fort Ross Virtual Warehouse serious game”. *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 231 – 238.
- Lipp, M., Wonka, P., Wimmer, M. (2008), “Interactive visual editing of grammars for procedural architecture”. In *SIGGRAPH'08: Proceedings of the 35th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA, 2008)*, ACM, pp. 1–10.
- Longay, S., Runions A., Boudon F., Prusinkiewicz P. (2012), “Treesketch: Interactive procedural modeling of trees on a tablet”. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (Aire-la-Ville, Switzerland, 2012)*, *Eurographics Association*, pp. 107–120.
- Macias, S. (1996a), “Mértola Islâmica: Estudo Histórico-Arqueológico do Bairro da Alcáçova, Séculos XII-XIII”. *Campo Arqueológico de Mértola*.

- Macias, S (1996b), "Islamic quarter", acessido a 30 de maio de 2016, em: http://www.discoverislamicart.org/database_item.php?id=monument;ISL;pt;Mon01;9;en.
- Magnenat-Thalmann, N., *et al.* (2006), "Real Time Animation of Ancient Roman Sites". *Computer Graphics and Interactive Techniques in Australasia and Southeast Asia: Proceedings of the 4th International Conference*, 2006, pp. 19 – 30.
- Martínez, S., *et al.* (2009), "XXII Mértola Islâmica. A madina e o arrabalde". Campo Arqueológico de Mértola, acessido a 13 de junho de 2016, em: <https://comum.rcaap.pt/handle/10400.26/2130>
- McHenry, K., Bajcsy, P. (2008), "An Overview of 3D Data Content, File Formats and Viewers". *ISDA - Innovative Software and Data Analysis Division*.
- Merlo, A., Fantini, F. (2012), "Game engine for Cultural Heritage: New opportunities in the relation between simplified models and database". *Virtual Systems and Multimedia (VSMM)*, 2012, 18th International Conference, pp. 623 – 628.
- Micoli, L., *et al.* (2013), "A multidisciplinary approach to 3D survey and reconstruction of historical buildings". *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 241 – 248.
- Musialski, P., *et al.* (2013), "A Survey of Urban Reconstruction". *Computer Graphics Forum*, Vol. 32, n.º. 6, pp. 146 – 177.
- Muller, P., *et al.* (2006), "Procedural Modeling of Buildings". *ACM Transactions On Graphics*, 2006, Jul, Vol. 25, pp. 614 – 623.
- Muller, T. (2012), "Mastering Blender". *John Wiley & Sons Inc.*, 2012.
- Papagiannakis, G., *et al.* (2002), "LIFEPLUS: Revival of life in ancient Pompeii". *Infoscience*, 2002.
- Parish Y. I. H. & Muller P. (2001), "Procedural modeling of cities". In *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), ACM, pp. 301–308.
- Patrício, P., *et al.* (2015), "Acesso à informação geográfica através do Sistema Nacional de Informação Geográfica e da iniciativa iGEO", acessido a 31 de março de 2016, em: http://snig.dgterritorio.pt/Inspire/documentos/VIIICNCG/CNCG_Patricio.pdf.
- Patow, G. (2012), "User Friendly Graph Editing Procedural Modeling of Buildings". *IEEE Computer Graphics and Applications*, March-April 2012, Vol. 32, pp. 66 – 67.
- Pietrone, E., *et al.* (2013), "Tiber Valley Virtual Museum: 3D landscape reconstruction in the Orientalising period, North of Rome: A methodological approach proposal". *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 223 – 230.
- Prusinkiewicz P. & Hammel, M. (1993), "A fractal model of mountains with rivers". In *Proceedings of Graphics Interface*, 1993, pp. 174–180.
- Rao, M. & Thakur, P. (2013), "Reconstruction of Virupaksha Bazaar Street of Hampi". *Digital Heritage International Congress (digitalHeritage)*, 2013, Vol. 2, pp. 207 – 214.
- Rodrigues, N., *et al.* (2008), "Automatic Reconstruction of Virtual Heritage Site". The 9th International Symposium on Virtual Reality, Archeology and Cultural Heritage.
- Rodrigues, N., *et al.* (2014), "Reconstruction and Generation of Virtual Heritage Sites". *Digital Applications in Archaeology and Cultural Heritage*, 2014, Vol. 1(3-4), pp. 92 – 102.

- Sabir, A. (2012), "Plug-in Development with C++ and the 3D's Max SDK". *Thesis of Master*, University Of Teesside, Middlesbrough (United Kingdom).
- Saldaña, M. & Johanson, C. (2013), "Procedural modeling for rapid-prototyping of multiple building phases". *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Fev., Vol. XL-5/W1.
- Santos, J. & Barata, J. (2013), "3ds Max Curso Completo" (2ª ed.). Lisboa: Editora FCA – Editora de informática, Ltda, 2013.
- Smelik, R., *et al.* (2014), "A Survey on Procedural Modelling for Virtual Worlds". *Computer Graphics Forum*, 2014, Vol. 33, num. 6, pp. 31 – 50.
- Stone, R. (2015), "Keynote Paper: virtual & augmented reality technologies for applications in cultural heritage: a human factors perspective".
- Torres, C. (2008), "Mértola: On the Mediterranean Commercial Routes". Câmara Municipal de Mértola, 2008.
- Troiano, D., *et al.* (2014), "From a Model of a City to an Urban Information System: The SIUR 3D of the Castle of Pietrabuona". *Springer International Publishing Switzerland*, 2014, pp. 121 – 130.
- Vlahakis, W., *et al.* (2012), "Archeoguide: An Augmented Reality Guide for Archaeological Sites". *Computer Graphics and Applications*, IEEE, Vol. 22, pp. 52 – 60.
- Zhou, H., *et al.* (2007), "Terrain Synthesis from Digital Elevation Models". *Jornal IEEE Transactions on Visualization and Computer Graphics*, Vol. 13. pp. 834-848.